

Microsoft BizTalk Adapter Pack 2010

Copyright© 2017 Microsoft Corporation

The content in this document is retired and is no longer updated or supported. Some links might not work. Retired content represents the latest updated version of this content.

Microsoft BizTalk Adapter for Oracle Database Help

Welcome to Microsoft BizTalk Adapter for Oracle Database Help, which includes a variety of resources that can help you learn to develop, deploy, administer, and use the Oracle Database adapter.

To download a copy of this Help, go to <http://go.microsoft.com/fwlink/?LinkID=188849>.

	Getting Started Information about new features, limitations, prerequisites, and topics for users who are new to the Oracle Database adapter.
	Oracle Database Adapter Migration Tutorial Step-by-step instructions for learning how to use the Oracle Database adapter for specific scenarios.
	Planning and Architecture Architectural overview of the Oracle Database adapter.
	Development Instructions on how to use the Oracle Database adapter with BizTalk Server, the Windows Communication Foundation (WCF) service model, and the WCF channel model.
	Deployment Information about installing the Oracle Database adapter, migrating from previous versions of the adapter, and troubleshooting the installation.
	Troubleshooting Information about enabling tracing and about troubleshooting issues with the BizTalk Adapter Pack and with the Oracle Database adapter, in particular.
	Samples Samples demonstrating how to use the Oracle Database adapter.
	Community Resources Information about how to get assistance with the BizTalk Adapter Pack.

Getting Started

This section provides an overview of the adapter, prerequisites, and topics for users who are new to Microsoft BizTalk Adapter Pack. It discusses the features of Microsoft BizTalk Adapter for Oracle Database and the different operations that can be performed on the Oracle database using the adapter.

What is an adapter? An adapter is a software component that enables you to send and receive messages to and from a line-of-business (LOB) system. The primary design goal of adapters is to facilitate the exchange of business documents between trading partners. Because each business system can adhere to specific document formats and protocols, the adapter uses a delivery mechanism that conforms to commonly recognized standards and protocols.

The adapters can be divided into two broad categories:

- **LOB adapters.** Such adapters provide a service-oriented programming model to access LOB systems—for example, adapters for SAP or Siebel.
- **Data adapters.** Such adapters provide a service-oriented programming model to access databases—for example, an adapter for the Oracle database or SQL Server.

There are five adapters in the BizTalk Adapter Pack:

- Microsoft BizTalk Adapter for SQL Server (the SQL adapter)
- Microsoft BizTalk Adapter for Oracle E-Business Suite (the Oracle E-Business adapter)
- Microsoft BizTalk Adapter for Oracle Database (the Oracle Database adapter)
- Microsoft BizTalk Adapter for mySAP Business Suite (the SAP adapter), including .NET Framework Data Provider for mySAP Business Suite (Data Provider for SAP)
- Microsoft BizTalk Adapter for Siebel eBusiness Applications (the Siebel adapter), including .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel)

Note

The Siebel adapter is not available for 64-bit platforms.

If you do not already know how you want to use the Oracle Database adapter at your company, it is recommended that you start by exploring the features and functionality of the adapter described in [Understanding BizTalk Adapter for Oracle Database](#).

In This Section

- [Prerequisite Skills and Knowledge](#)
- [Common Administrative Tasks](#)
- [Common Developer Tasks](#)
- [Understanding BizTalk Adapter for Oracle Database](#)
- [Oracle Database Adapter Migration Tutorial](#)
- [Community Resources](#)
- [Frequently Asked Questions](#)

Prerequisite Skills and Knowledge

The potential users for the BizTalk Adapter Pack are:

- Developers who directly program to the adapters
- IT professionals who consume the adapters using other integration platforms like Microsoft BizTalk Server and other tools.
- Independent software vendors (ISVs) who build solutions on top of the adapters

The prerequisite skills and knowledge required for each role are detailed below.

Prerequisites for Developers

A developer using Microsoft BizTalk Adapter for Oracle Database must be at least moderately experienced with the use of:

- Microsoft Visual Studio 2010 , with the development of .NET solutions
- Programming with the .NET Framework 4 or .NET Framework 3.5 SP1
- Programming with the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
- Extensible Markup Language (XML)
- XML Schema definition (XSD) language
- Web Services Description Language (WSDL)

Prerequisites for IT Professionals

An IT professional using the Oracle Database adapter must be at least moderately experienced with the use of:

- SQL Server Integration Services (SSIS)
- BizTalk Server 2010
- Windows SharePoint Services

Prerequisites for ISVs

ISVs using the Oracle Database adapter must be at least moderately experienced with:

- The internal workings and concepts of the Oracle database to be able to build applications on top of the adapters
- The .NET Framework 4 or .NET Framework 3.5 SP1
- The WCF LOB Adapter SDK

See Also

Other Resources

[Getting Started](#)

[Understanding BizTalk Adapter for Oracle Database](#)

Common Administrative Tasks

The following table provides links to topics helpful for deploying, using, and maintaining Microsoft BizTalk Adapter for Oracle Database.

Administrative Tasks <ul style="list-style-type: none">• Oracle Database Adapter Migration Tutorial• Deployment	Planning and Architecture <ul style="list-style-type: none">• Architecture for BizTalk Adapter for Oracle Database• Understanding BizTalk Adapter for Oracle Database• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Glossary
---	---

See Also

Concepts

[Common Developer Tasks](#)

Other Resources

[Getting Started](#)

Common Developer Tasks

The following table provides links to topics useful for developing applications on top of Microsoft BizTalk Adapter for Oracle Database.

Developer Tasks <ul style="list-style-type: none">• Establishing a Connection to the Oracle Database• Developing BizTalk Applications• Developing Applications by Using the WCF Service Model• Developing Applications by Using the WCF Channel Model	Planning and Architecture <ul style="list-style-type: none">• Architecture for BizTalk Adapter for Oracle Database• Understanding BizTalk Adapter for Oracle Database• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Troubleshooting• Technical Reference• Glossary
---	---

See Also

Concepts

[Common Administrative Tasks](#)

Other Resources

[Getting Started](#)

Understanding BizTalk Adapter for Oracle Database

The BizTalk Adapter Pack enables service-oriented programmatic access in order to interact with an external system. The adapters provide the following advantages to clients:

- **Consistent design-time experience.** The adapters provide a common and user-friendly design-time experience for browsing, searching, and retrieving metadata of LOB artifacts.
- **Varied programming options.** The adapters provide a choice of programming model including the Windows Communication Foundation (WCF) channel model, WCF service model, ADO.NET, Web services, or BizTalk supported models.
- **Uniform experience across LOBs.** The adapters standardize on using WCF and Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, and hence provide a uniform experience of gaining access to any LOB system.

As mentioned, the adapters are built on top of the WCF LOB Adapter SDK. The WCF LOB Adapter SDK provides a common basis for building integration adapters that a variety of client applications such as BizTalk Server and Microsoft Office can consume. The WCF LOB Adapter SDK aligns the adapter strategy with the Microsoft Services strategy by exposing integration adapters as Windows Communication Foundation (WCF) channels. For more information about the WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation. The WCF LOB Adapter SDK documentation is installed along with the WCF LOB Adapter SDK, typically under <installation drive>\Program Files\WCF LOB Adapter SDK\Documents.

To perform operations on an Oracle database, adapter clients must have access to relevant tables, functions, and procedures. Database tables are the basic unit of storage in the Oracle database. External applications can add or remove data from a table by using SQL statements. Applications can also access data in the tables by using views, functions, and procedures. With Microsoft BizTalk Adapter for Oracle Database, adapter clients can browse the artifacts such as tables, procedures, packages, views, and other such items in an Oracle database. Adapter clients can select the artifacts they require for their solution and retrieve metadata for those artifacts. This enables users to access and execute the operations on the artifacts in the Oracle database.

This section lists the features of the Oracle Database adapter.

In This Section

- [Overview of BizTalk Adapter for Oracle Database](#)
- [New and Deprecated Features in BizTalk Adapter for Oracle Database](#)
- [Limitations of BizTalk Adapter for Oracle Database](#)

See Also

Other Resources

[Getting Started](#)

Overview of BizTalk Adapter for Oracle Database

The Microsoft BizTalk Adapter for Oracle Database exposes the Oracle database as a WCF service. Adapter clients can perform operations on the Oracle database by exchanging SOAP messages with the adapter. The adapter consumes the WCF message and makes appropriate ODP.NET calls to perform the operation. The adapter returns the response from the Oracle database back to the client in the form of SOAP messages.

The Oracle Database adapter surfaces metadata of Oracle database artifacts (tables, functions, procedures, etc.) that describes the structure of a SOAP message in the form of WSDL. The Oracle Database adapter uses Add Adapter Service Reference Visual Studio Plug-in, Consume Adapter Service BizTalk Project Add-in, and Add Adapter Metadata Wizard to enable adapter clients to retrieve metadata for operations and generates programming artifacts that can be used in your programming solution.

The Oracle Database adapter uses the Oracle Data Provider for .NET (ODP.NET) 11.1.0.7 to communicate with the Oracle database. You can use the Oracle Database adapter to communicate with the Oracle database in the following ways:

- By developing BizTalk applications. See [Developing BizTalk Applications](#) for more information.
- By using the Windows Communication Foundation (WCF) service model. See [Developing Applications by Using the WCF Service Model](#) for more information.
- By using the WCF channel model. See [Developing Applications by Using the WCF Channel Model](#) for more information.

In This Section

- [How Does the Adapter Connect to an Oracle Database?](#)
- [How Does the Adapter Surface Oracle Metadata?](#)
- [What Operations Can be Performed Using the Adapter?](#)
- [How does the Adapter Handle Transactions?](#)
- [Streaming Support for LOB Data Types](#)
- [Other Features Supported by the Adapter](#)

See Also

Other Resources

[Understanding BizTalk Adapter for Oracle Database](#)

How Does the Adapter Connect to an Oracle Database?

The Microsoft BizTalk Adapter for Oracle Database uses ODP.NET 11.1.0.7 to connect to the Oracle database. The Oracle Database adapter requires adapter clients to provide a connection string, called the connection Uniform Resource Identifier (URI), to connect to the Oracle database. Internally, the Oracle Database adapter maps the URI to a database connection string to connect to the Oracle database. With a connection URI, adapter clients can specify connection parameters to connect to an external system.

The Oracle Database adapter enables adapter clients to connect to the Oracle database in the following two ways:

- **Using tnsnames.ora:** The connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, and port number from the net service name entry in the tnsnames.ora file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.

◆ Important

Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the [connection URI](#) cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

- **Without using tnsnames.ora:** The connection URI provided by the adapter clients contains the connection parameters such as server name, service name, and port number. In this case, the net service name in the tnsnames.ora file, or the actual tnsnames.ora file itself, does not need to be present on the client computer. This is helpful when you have a large number of users connecting to the Oracle database in your organization, and adding/updating servers does not lead to manually adding/updating the connection details in the tnsnames.ora file on every client computer.

◆ Important

This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

For more information about connecting to the Oracle database, see [Establishing a Connection to the Oracle Database](#).

Make sure you comply with the security guidelines when establishing a connection with the Oracle database. For more information about security guidelines, see [Security and Protection](#).

Windows Authentication

The Oracle Database adapter supports Windows Authentication while connecting to the Oracle database. With Windows Authentication, the adapter clients can determine a user's identity based on Windows logon credentials, and can leverage the built-in security of the Windows environment. For information about connecting to the Oracle database by using Windows Authentication, see [Connecting to the Oracle Database Using Windows Authentication](#).

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

How Does the Adapter Surface Oracle Metadata?

The Oracle Database adapter surfaces metadata from the Oracle database that describes the message structure for communicating with the Oracle database using the adapter. The Oracle Database adapter supports two interfaces for retrieving metadata.

- **MetadataExchange** provided by Windows Communication Foundation (WCF). WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from the Oracle database.
- **IMetadataRetrievalContract** provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

The Oracle Database adapter surfaces the Oracle database artifacts and respective operations that the adapter clients can invoke. The adapter also surfaces the **SQLEXECUTE**, **POLLINGSTMT**, and **Notification** operations that can be used to perform specific operations on the Oracle database. These operations are discussed later in this topic.

Adapter clients can browse, search, and retrieve metadata by using the WCF channel model, by using the WCF service model, or by creating a BizTalk project in Visual Studio. When using the WCF service model, you must use the **Add Adapter Service Reference Visual Studio Plug-in** to generate the proxy classes for performing operations on the Oracle database. When using a BizTalk project, you must use the **Consume Adapter Service BizTalk Project Add-in** or the **Add Adapter Metadata Wizard** to generate metadata for the operations that you want to perform on the Oracle database. For more information about browsing, searching, and retrieving metadata using **Add Adapter Service Reference Plug-in**, **Consume Adapter Service Add-in** or **Add Adapter Metadata Wizard**, see [Retrieving Metadata for Oracle Operations in Visual Studio](#).

Browsing Metadata

The Oracle Database adapter enables adapter clients to browse database tables, table views, stored procedures, functions, and packages that are available in the Oracle database. As part of the metadata browse operation, the adapter also surfaces the operations that can be performed on the Oracle database, including some custom operations supported by the adapters. These operations are available from the **Add Adapter Service Reference Plug-in** or the **Consume Adapter Service Add-in**. The Oracle Database adapter surfaces the following operations:

Outbound Operations

Contains a list of schemas in the underlying Oracle database. Expand a schema node to see the following artifacts:

- **Table:** A list of all the tables in the schema. Select a table to view the Insert, Select, Update, and Delete operations.
- **Procedure:** A list of stored procedures in the schema that are exposed as operations.
- **Function:** A list of functions in the schema that are exposed as operations.
- **Package:** A list of all the packages in the schema. Select a package to view the procedures and functions inside the package that are exposed as operations.
- **View:** A list of all the views in the schema. Select a view to view the Insert, Select, Update, and Delete operations.

Apart from this, the Oracle Database adapter also exposes the **SQLEXECUTE** outbound operation, which enables the adapter clients to execute any generic data manipulation language (DML) or stored procedure in an Oracle database. The **SQLEXECUTE** operation is available when you select the root node (/). Note that the output of **SQLEXECUTE** is an array of data readers (output as array of generic records). As a result, any simple out parameters are not surfaced using the **SQLEXECUTE** operation. For more information about the operation, see [SQLEXECUTE Operation](#).

Inbound Operations

Contains a list of schemas in the underlying Oracle database. Expand a schema node to see the following artifacts:

- **Procedure:** A list of stored procedures in the schema that are exposed as operations for polling.
- **Function:** A list of functions in the schema that are exposed as operations for polling.

- **Package:** A list of packages in the schema. Select a package to view the packaged procedures and functions that are exposed as operations for polling.

Apart from this, the Oracle Database adapter also exposes the **POLLINGSTMT** and **Notification** inbound operations. The POLLINGSTMT operation enables adapter clients to obtain inbound data from the Oracle database based on a query polling mechanism supported by the adapter. The Notification operation enables adapter clients to register a SELECT statement as the notification query on the database, and the database sends a notification to the adapter client as and when the result set of the SELECT statement changes. The POLLINGSTMT and Notification operations are available when you select the root node (/). For more information about the operations, see [Support for Receiving Polling-based Data-changed Messages](#) and [Receiving Database Change Notifications](#).

For more information about how the metadata is categorized, see [Metadata Node IDs](#).

Searching Metadata

With the Oracle Database adapter, it is possible to perform a search query on the Oracle database by using the Oracle search expressions that are compatible with the LIKE operator. For example, adapter clients can use a search expression such as "EMP%" to obtain tables starting with EMP. The adapter converts this to the following SQL query:

```
SELECT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME LIKE 'EMP%' AND OWNER = 'SCOTT'
```

Where, SCOTT is the schema with a collection of Oracle database artifacts.

The following table lists the special characters that can be used for search and their interpretation by the Oracle Database adapter.

Special character	Interpretation
_ (underscore)	Matches exactly one character For example, A_ matches AB, AC, AD.
% (percentage)	Matches zero or more characters. For example, A% matches A, AB, ABC.
\ (escape)	Escapes the special meaning of % and _ For example, A_B matches A_B.

◆ Important

The metadata search scope is restricted to the level immediately under the node at which the search operation is performed. For example, to search for a function, you must be searching under \[Schema]\Functions. Recursive search is not supported.

Retrieving Metadata

When retrieving metadata, the Oracle Database adapter can extract metadata under a schema, including all or a subset of database objects with the respective object and operation parameters. The adapter presents the entities from the Oracle database as element names in XML. Because underscores are the only permissible special characters that can be included, all other special characters in the element names are encoded using underscores. For example, emp\$name is encoded as emp_x0024_name.

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

Other Resources

[Understanding BizTalk Adapter for Oracle Database](#)

[Retrieving Metadata for Oracle Operations in Visual Studio](#)

What Operations Can be Performed Using the Adapter?

Adapter clients can perform operations on the Oracle database by creating BizTalk projects, by using the WCF channel model, or by using the WCF service model. The Oracle Database adapter exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel. For information about the message structure and the SOAP action associated with each operation, see [Messages and Message Schemas for BizTalk Adapter for Oracle Database](#).

This section provides information about the operations supported on the Oracle database using the Oracle Database adapter.

In This Section

- [Performing Basic Insert, Update, Delete, and Select Operations on Oracle Tables and Views](#)
- [Operations on Tables and Views That Contain LOB Data](#)
- [Operations on Functions and Stored Procedures](#)
- [Operations on Functions and Procedures with REF CURSOR Parameters](#)
- [Operations on Functions and Procedures with RECORD Types](#)
- [Operations on Tables With BFILE Data Types](#)
- [Operations on Synonyms](#)
- [SQLEXECUTE Operation](#)
- [Performing Composite Operations](#)
- [Support for Receiving Polling-based Data-changed Messages](#)
- [Receiving Database Change Notifications](#)
- [Support for Oracle User-Defined Types](#)

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

Performing Basic Insert, Update, Delete, and Select Operations on Oracle Tables and Views

The Microsoft BizTalk Adapter for Oracle Database surfaces a set of standard operations on each Oracle database table and view. By using these operations, you can perform simple SQL INSERT, UPDATE, SELECT, and DELETE statements qualified by a WHERE clause on the target table (or view). These operations are also called data manipulation language (DML) operations. To perform more complex operations, for example a SQL SELECT query that uses the JOIN operator, you can use the SQLEXECUTE operation. For more information about the SQLEXECUTE operation, see [SQLEXECUTE Operation](#).

The following table shows the DML operations that the Oracle Database adapter supports:

Operation	Description
-----------	-------------

Insert Performs an Insert operation on the target table or view. The Insert operation supports multiple record or bulk inserts into the target table or view:

- A multiple record Insert operation inserts rows into a table or view based on a supplied record set.
- A bulk Insert operation inserts rows into a table or view based on a supplied SQL SELECT query and column list. The records that the query returns are inserted into the target table based on the column list.

The return value for an Insert operation is the number of rows inserted.

 **Note**

Both multiple-record insert and bulk insert cannot be combined in the same message.

InlineValue

For all simple data records in a multiple record Insert operation, you can choose to override the value of a record by specifying a value for an optional attribute called **InlineValue**. The InlineValue attribute can be used to insert computed values into tables or views such as populating the primary key column using a sequence or inserting system date (using SYSDATE) into a date column. For example, in the following INSERT statement:

```
<Insert xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY" >
  <RECORDSET>
    <ACCOUNTACTIVITYRECORDINSERT>
      <ACCOUNT>10001</ACCOUNT>
      <EMPNAME>John</EMPNAME>
      <AMOUNT>1500</AMOUNT>
      <TRANSDATE InlineValue="SYSDATE">2008-06-21T15:52:19</TRANSDATE>
    </ACCOUNTACTIVITYRECORDINSERT >
  </RECORDSET>
</Insert>
```

Even though "2008-06-21T15:52:19" is specified as a value for the TRANSDATE column, the value of the InlineValue attribute, "SYSDATE," (system date) will be inserted into the target table.

While using the InlineValue attribute:

- Avoid using constant values for the InlineValue attribute. For example, in the INSERT statement, if you specify `<EMPNAME InlineValue="John"/>` then it will result in an error. This is because the value of the InlineValue attribute is passed on as-is to Oracle, and in this case *John* is passed to the Oracle database, which is not the expected value (expected value is *John*). You would have to use single quotes around the employee name. For example: `<EMPNAME InlineValue="'John'"/>`.
- If you want to use a select query for the InlineValue attribute, you must enclose the SELECT statement in parentheses and also ensure that the select query fetches only a single record. For example: `<EMPNAME InlineValue="(SELECT NAME FROM MS_SAMPLE_EMPLOYEES WHERE ID=123)"/>`.

 **Note**

If an element is marked as NOT NULL in the Oracle database, you must specify a value for that element even if you have specified an inline value. Failing to do this will cause the schema validation to fail.

Select Performs a SQL SELECT query on the target table or view based on a supplied list of column names and a filter string that specifies a SQL WHERE clause.

The return value for a Select operation is a strongly-typed result set that contains the specified columns and rows.

Update	<p>Performs an Update operation on the target table or view. The records to be updated are specified by a filter string that specifies a SQL WHERE clause. The values for the update are specified in a template record.</p> <p>The return value for an Update operation is the number of rows updated.</p>
Delete	<p>Performs a Delete operation on the target table or view based on a SQL WHERE clause that is specified in a filter string.</p> <p>The return value for a Delete operation is the number of rows deleted.</p>

For more information about:

- Performing these operations using BizTalk Server, see [Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server](#).
- Performing these operations using the WCF service model, see [Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model](#).
- Performing these operations using the WCF channel model, see [Performing an Insert Operation by Using the WCF Channel Model](#).
- Message structures and SOAP action for performing DML operations, see [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Tables and Views That Contain LOB Data

The Microsoft BizTalk Adapter for Oracle Database provides support for the Oracle large object (LOB) data types:

- Binary large object (BLOB)
- Character large object (CLOB)
- National character large object (NCLOB)
- Binary file (BFILE). For more information, see [Operations on Tables With BFILE Data Types](#).

On the Oracle database, LOB data types are used to store large amounts of data (up to 4 GB). LOB types support both input and output streaming.

The Microsoft BizTalk Adapter for Oracle Database surfaces the following operations for tables and views that contain LOB columns:

- **ReadLOB.** The ReadLOB operation is surfaced for tables and views that contain BLOB, CLOB, NCLOB, and BFILE columns. By using the ReadLOB operation, adapter clients can read values in a LOB column as a data stream. This operation takes the LOB data type column name and a filter string as parameters. Adapter clients must ensure that the filter string fetches exactly one matching row. If there is more than one matching row, the Oracle Database adapter only returns the LOB column for the first (matching) row.

Note

The ReadLOB operation is designed to support input streaming of LOB data in the WCF service model. You should use a table Select operation to read LOB data from a WCF Channel Model or BizTalk Server solution. For more information about streaming, see [Streaming Support for LOB Data Types](#).

- **UpdateLOB.** The UpdateLOB operation is surfaced for tables and views that contain BLOB, CLOB, and NCLOB columns. By using the UpdateLOB operation, adapter clients can update values in a LOB column. This operation takes the LOB data type column name, a filter string, and base64binary encoded data as parameters. Adapter clients must ensure that the filter string fetches exactly one matching row; otherwise the Oracle Database adapter throws an XmlReaderParsingException.

Note

The UpdateLOB operation:

- Is not supported for the BFILE data type. Adapter clients can alternatively use the Update operation. For more information, see [Operations on Tables With BFILE Data Types](#).
- Must be performed as part of a transaction. To ensure this, the **UseAmbientTransaction** binding property must be set to **True**. For information about the **UseAmbientTransaction** binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

ReadLOB and UpdateLOB operate on a single LOB column in a single table row. To operate on LOB columns in multiple rows or on multiple LOB columns within a single row, you must invoke ReadLOB or UpdateLOB for each target column within each target row.

For more information about:

- Invoking the UpdateLOB operation on an Oracle database table using BizTalk Server, see

[Performing Operations on Tables with Large Object Types Data by Using BizTalk Server](#). (You should use a table Select operation to read LOB data types in BizTalk Server.)

- Invoking ReadLOB and UpdateLOB operations on an Oracle database table using WCF service model, see [Performing Operations on Tables with Large Object Types by Using the WCF Service Model](#).
- Message structure and SOAP actions for performing ReadLOB and UpdateLOB operations, see [Message Schemas for Special LOB Operations](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Functions and Stored Procedures

The Microsoft BizTalk Adapter for Oracle Database supports Oracle functions, procedures, and packages in the following manner:

- **Functions** are surfaced as operations. The name of the operation is the name of the Oracle function. IN, OUT, and IN OUT parameters are supported, as well as, RETURN values.
- **Procedures** are surfaced as operations. The name of the operation is the name of the Oracle procedure. IN, OUT, and IN OUT parameters are supported.
- **Packaged functions and procedures** are surfaced as operations. The name and namespace of the operation (function or procedure) is qualified by the name of the Oracle package. Overloads are supported in packages (see next bullet).
- **Overloaded functions and procedures** in packages are surfaced as operations. Each overloaded function or procedure is surfaced with a string appended to its name that identifies the overload. This string is part of the sequence "overload1", "overload2", "overload3", and so on.
- **REF CURSOR types** are supported for IN, OUT, and IN OUT parameters for procedures and functions, as well as for function RETURN values. For more information, see [Operations on Functions and Procedures with REF CURSOR Parameters](#).
- **RECORD types** are supported for IN, OUT, and IN OUT parameters for procedures and functions, as well as for function RETURN values. Both simple and complex (nested) RECORD types are supported. [Operations on Functions and Procedures with RECORD Types](#)

For more information about:

- Invoking an Oracle procedure or function by using BizTalk Server, see [Invoking Functions and Procedures by Using BizTalk Server](#) and [Invoking Overloaded Functions and Procedures by Using BizTalk Server](#).
- Invoking an Oracle procedure or function by using the WCF service model, see [Invoking Functions and Procedures by Using the WCF Service Model](#).
- Invoking an Oracle procedure or function by using the WCF channel model, see [Invoking a Function by Using the WCF Channel Model](#).
- Message structure and SOAP actions used to invoke Oracle procedures and functions, see [Message Schemas for Functions and Procedures](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Functions and Procedures with REF CURSOR Parameters

A REF CURSOR is a PL/SQL data type that represents a pointer to a server-side result set generated by executing a query. A REF CURSOR type enables input and output streaming of data and is ideal for transferring large amounts of data to and from a PL/SQL code. The Oracle Database adapter provides support for strongly-typed and weakly-typed (SYS_REFCURSOR) REF CURSORS that can be passed to PL/SQL procedures and functions as IN, OUT, or IN OUT parameters.

- **IN REF CURSOR.** Adapter clients must use an IN REF CURSOR by supplying a PL/SQL code (as string) that opens REF CURSOR on the Oracle database. The adapter creates variable and sets it the opened REF CURSOR and calls a function or procedure with that variable. Therefore, IN REF CURSOR parameters in PL/SQL stored procedure and functions should be represented as strings that take a PL/SQL code block as input value marking the OUT REF CURSOR variable with a "?".
- **OUT REF CURSOR.** OUT REF CURSOR parameters are returned as either strongly-typed or weakly-typed result sets. The type of the result set returned depends on whether the REF CURSOR parameter is declared as a strongly-typed or weakly-typed REF CURSOR in the stored procedure or function definition on the Oracle server.
- **IN OUT REF CURSOR parameters.** Because the Oracle Database adapter models IN REF CURSOR parameters as strings and OUT REF CURSOR parameters as complex types, it cannot support a single type for an IN OUT REF CURSOR parameter. For this reason, it treats IN OUT REF CURSOR parameters as two different parameters: an IN parameter in the request message and an OUT parameter in the response message.

For more information about:

- Invoking a function or procedure involving REF CURSOR parameters using the BizTalk Server, see [Invoking Functions and Procedures with REF CURSORS by Using BizTalk Server](#).
- Invoking a function or procedure involving REF CURSOR parameters using the WCF service model, see [Performing Operations Using REF CURSORS in the WCF Service Model](#).
- XML structure for REF CURSORS as supported by the Oracle Database adapter, see [Message Schemas for REF CURSORS](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Functions and Procedures with RECORD Types

Oracle RECORD types are used to represent hierarchical information in parameters passed to PL/SQL functions and procedures. The Microsoft BizTalk Adapter for Oracle Database surfaces RECORD types as complex XML types. The Oracle Database adapter supports the following kinds of RECORD types:

- RECORD types that are declared as TABLE%ROWTYPE parameters in stored procedures and functions.
- RECORD types that are declared as TYPE of RECORD parameters in PL/SQL packages for example, TYPE rec_type1 IS RECORD(name varchar2(100), age number(3));
- RECORD types that contain nested records.
- RECORD types that appear as IN, OUT, or IN OUT parameters to procedures or functions.
- RECORD types that are RETURN values of functions.

Note

The Oracle Database adapter does not support BFILE types as RECORD members.

For information about:

- Invoking a function or procedure involving RECORD types using the WCF service model, see [Performing Operations Using RECORD Types in the WCF Service Model](#).
- Invoking a function or procedure involving RECORD types using BizTalk Server, see [Invoking Functions and Procedures with RECORD Types by Using BizTalk Server](#).
- XML structure for RECORD types as supported by the Oracle Database adapter, see [Message Schemas for RECORD Types](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Tables With BFILE Data Types

The Oracle Database adapter supports the BFILE data type in tables and stored procedures. The following table summarizes the BFILE data type exposed by the adapter based on the operation performed and the LOB artifact (table/procedure) accessed:

Artifact	Operation	Data type exposed for BFILE column/param	Comments
TABLE	INSERT	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
TABLE	UPDATE	String	Represents the logical Oracle directory path to the file to be updated into the BFILE column
TABLE	SELECT	byte[]	Represents the binary data constituting the BFILE
STORED PROC	IN PARAM	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
STORED PROC	OUT PARAM	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
STORED PROC	INOUT PARAM	Not Supported	-

The special operation ReadLOB is also supported on tables with BFILE data type. The UpdateLOB operation is not supported. Adapter clients can alternately use the UPDATE operation.

For more information about:

- Performing operations on tables containing BFILE data types by using BizTalk Server, see [Performing Operations on Tables with BFILE Data Types by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Synonyms

The Oracle Database adapter allows you to perform operations on synonyms. A synonym is an alias or friendly name for the database objects (such as tables, views, stored procedures, functions, and packages). For more information about synonyms in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=138058>.

Advantages of Using Synonyms

Synonyms are helpful in the following scenarios:

- **Working with different schemas:** If you are working with different schemas, and need to access the objects across schemas, you have to use different SQL statements to access those objects. You can create a synonym for an object in a schema, and use the synonym in your SQL statement to access the object. If you need to access the underlying object in a different schema, modify the definition of the synonym to point to the object in a different schema. Thus, the applications based on the synonym continue to function without modification in the SQL statement.

For example, suppose you have two identical schemas for your test and production environments: "Test" and "Prod." To access a table called "Employee" in the "Test" schema, you must use `Test.Employee` or `Employee` (if "Test" is the default schema) in your SQL statement. If you want to use the "Employee" table in the production schema, you must now use `Prod.Employee` or `Employee` (change the default schema to "Prod") in your SQL statement. To get around this issue, you can create a synonym for the "Test.Employee" table (say "EMP"), and then use it in your SQL statements. Whenever you need to perform operation on the "Prod.Employee" table, modify the definition of the "EMP" synonym to point it to the "Prod.Employee" table. This ensures that you do not have to modify your SQL statements to perform operation on the object in different schemas.

- **Changes in the underlying objects:** The synonyms insulate you from any changes in the name or location of the underlying objects on which you are performing an operation. You can modify the synonym definition to accommodate any changes in the name or location of the underlying objects.

For example, suppose that you are using a table in one of your stored procedures. Now, if the table name changes or the table is moved to some other location then your stored procedure will stop working. To work around this, you can use a synonym for the table in the stored procedure, and update the synonym definition if there is a change in the name or location of the table.

- **Simplified and secure access:** In a distributed environment, you must use the schema name along with the object names to ensure that you are accessing the correct object. Moreover, you must also ensure that the user has required privileges on the target object. To simplify this, you can assign a simple name for an object by creating a synonym that has the full qualified path to the object, and then grant appropriate privileges on the synonym.

Working with Synonyms in the Adapter

The Oracle Database adapter exposes the synonyms in Oracle for:

- Tables
- Views
- Stored Procedures
- Functions
- Packages

The synonyms for each of these artifacts are exposed alongside the respective underlying artifact in the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Plug-in. For example, the **Table** node under a schema will display all the synonyms for tables along with the database tables in a schema, the **View** node under a schema will display all the synonyms for views along with the database views in a schema, and so on.

- For synonyms created on tables and views, the same operations are exposed as for the underlying tables and views respectively. For example, if the underlying tables and views contain LOB columns, the synonyms for those tables and views will also expose the ReadLOB and UpdateLOB operations.
- For synonyms created on stored procedures, functions, and packages, the synonyms are exposed as operations alongside the respective underlying stored procedures, functions, and packages in a schema.

Note

The Oracle Database adapter supports only local synonyms. This implies that only those synonyms are supported by the adapter that target the artifacts on the local server.

Moreover, the message actions for the synonyms are the same as the underlying object except for the artifact name on which the action is performed. For example, the message action for the **Select** operation on a table in the SCOTT schema is:

`http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/[TABLE_NAME]/Select`. If you are performing a Select operation on a synonym for the same table in the SCOTT schema then the message action will be:

`http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/[SYNONYM_NAME]/Select`.

When you invoke an operation on a synonym in the adapter, the adapter calls the synonym in the Oracle database to execute the operation. However, the adapter uses the underlying object name in the synonym definition to fetch the metadata.

Synonyms can be used in normal outbound operations, composite operations, and polling.

Note

You can search for synonyms in Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in just like other objects. However, you cannot search for procedures inside synonym packages from a skip-level node as you can do for the procedures inside packages. For information about searching for operations in the adapter, see [Browsing, Searching, and Retrieving Metadata for Oracle Operations](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

SQLEXECUTE Operation

The Microsoft BizTalk Adapter for Oracle Database surfaces a standard set of operations on Oracle database artifacts. By using these operations, you can do things like call an Oracle function or procedure, or perform basic SQL data manipulation language (DML) operations on tables. However, there may be scenarios driven by your business logic that require you to perform operations that the Oracle Database adapter does not surface. For example, you may want to:

- Perform an operation on database artifacts that are not surfaced by the Oracle Database adapter; for example, get the CURVAL or NEXTVAL of an Oracle SEQUENCE.
- Perform data definition language operations; for example, create a table.
- Perform operations on a database artifact that was not present at design time; for example, update records in a temporary table that is created by your business logic.
- Perform more complex DML operations on tables than the operations that the Oracle Database adapter surfaces; for example, to perform a query that includes a JOIN clause.

For these kinds of scenarios, the Oracle Database adapter surfaces the SQLEXECUTE operation. The SQLEXECUTE operation is surfaced under the root node (/) in the **Select a category** pane in the Add Adapter Service Reference Visual Studio Plug-in and Consume Adapter Service BizTalk Project Add-in.

By using the SQLEXECUTE operation, you can perform a parameterized SQL statement on the Oracle database. The SQLEXECUTE operation supports an input parameter block consisting of parameter sets that enable you to execute the same SQL statement once for each set. The SQLEXECUTE operation returns the results of the SQL statement in a generic record set.

Note

You can pass IN and IN OUT parameters to procedures, functions, and packages in the SQLEXECUTE operation. The invoked artifact will execute with the supplied parameters on the Oracle database; however, the SQLEXECUTE operation does not return the value of OUT and IN OUT parameters to the client. If you want to invoke procedures, functions, or packages, we recommend that you do so by invoking the dedicated operations that the Oracle Database adapter exposes for these Oracle artifacts.

For more information about:

- Performing a SQLEXECUTE operation by using BizTalk Server, see [Performing a SQLEXECUTE Operation by Using BizTalk Server](#).
- Performing a SQLEXECUTE operation by using the WCF service model, see [Performing a SQLEXECUTE Operation by Using the WCF Service Model](#).
- Performing the SQLEXECUTE operation by using the WCF channel model, see [Performing a SQLEXECUTE Operation by Using the WCF Channel Model](#).
- Message structure and SOAP actions for performing a SQLEXECUTE operation, see [Message Schemas for the SQLEXECUTE Operation](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Performing Composite Operations

The Oracle Database adapter enables adapter clients to perform composite operations that can include any number of the following operations, and in any order:

- Select, Insert, Update, and Delete operations on tables and views.
- Stored procedures, functions, and procedures or functions within packages that are surfaced as operations in the adapter.

The operations in a composite operation can target tables and views in the same database or different databases. However, data cannot be shared or reused across different operations in a composite operation. For example, in a composite operation, the result set of a Select operation cannot be used as the input parameter for a stored procedure.

Each operation in a composite operation is performed using a separate connection. The Oracle Database adapter consumes as many connections from the ODP.NET connection pool as the number of operations in a composite operation, and then releases the connections as the operations get executed. However, if an operation in the composite operation returns a result set, the connection is released only after the message is consumed.

◆ Important

If you experience time-out issues while executing a composite operation then it could be because the number of connections is less than the number of operations in a composite operation involving:

- Stored procedures containing BFILE, BLOB, CLOB, NCLOB, and REF CURSOR as OUT or IN OUT parameters.
- Select operation.

To resolve this issue, you must ensure that if there are "n" number of such operations in a composite operation, the value specified for the **MinPoolSize** binding property is "n+1" or greater. For more information about the **MinPoolSize** binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

For information about:

- How to perform composite operations in Oracle Database adapter using BizTalk Server, see [Performing Composite Operations on Oracle Database by Using BizTalk Server](#).
- Message schemas for the composite operation, see [Message Schemas for the Composite Operation](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Support for Receiving Polling-based Data-changed Messages

The Oracle Database adapter enables client programs to receive messages from the Oracle database informing them of changes to data stored in an Oracle database. The Oracle Database adapter supports receiving "polling-based" messages wherein the adapter executes a specified SELECT query, stored procedure, function, or procedure or function within a package, retrieves the data, and provides the result to the client at regular intervals of time. To enable this, the Oracle Database adapter exposes a POLLINGSTMT operation. Moreover, all the stored procedures, functions, and procedures and function within packages are exposed as inbound operations for polling.

The adapter provides two ways of polling the Oracle database:

- **Using SELECT statements.** You can specify a simple SELECT statement to poll the tables and views in the Oracle database. The adapter executes the SELECT statement at specified intervals and returns the result to the adapter clients.
- **Using stored procedures, functions, or procedures or functions within a package.** You can specify a stored procedure, function, or procedure or function within a package to poll the Oracle database. The adapter executes the request message at specified intervals and returns the result to the adapter clients.

Polling Operation Workflow

A typical polling operation using the Oracle Database adapter involves the following:

1. The adapter clients must specify **Polling** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is **Polling**.
2. The adapter clients must specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine whether there is data available for polling. On execution of this statement, if the first column of the first row of the result set returned contains a positive integer value, there is data available for polling. By default, the value of this binding property is set to `Select 1 FROM DUAL`, which implies that the adapter must continue polling irrespective of whether the table being polled has data or not.
3. The adapter clients must specify a polling interval for the **PollingInterval** binding property to define the interval in seconds at which the statement specified in the **PolledDataAvailableStatement** binding property is executed. At the end of every polling interval, the polled data available statement is executed, and the result set is returned.
4. The adapter clients must specify a SELECT statement or a stored procedure for the **PollingStatement** binding property.
 - If you want to poll a table or view, you must specify a SELECT query in this binding property.
 - If you want to poll using a stored procedure, function, or procedure or function within a package, you must specify the entire request message for the respective operation in this binding property.

The statement in the **PollingStatement** binding property is executed only if there is data available for polling, which is determined by the **PolledDataAvailableStatement** binding property in step 1.

5. The adapter clients must specify an action for the polling operation in the **PollingAction** binding property. The polling action for a specific operation is determined from the metadata generated for the operation using the Consume Adapter Service Add-in.

Note

If you are polling a table or view using a SELECT statement in the **PollingStatement** binding property, you do not need to specify any value for the **PollingAction** binding property. The default value, Null, is passed in this case.

6. The adapter clients can use the **PollWhileDataFound** binding property to ignore the polling interval, and continuously poll data, as and when available.

◆ Important

If you set the value of the **PollWhileDataFound** binding property to True, the adapter clients continuously poll data from Oracle and in the process open and close connections to the Oracle database in a loop. As the rate at which connections are opened by ODP.NET is greater than the connections being closed, the connections get exhausted after some time, and an exception is thrown. As a work around, make sure that the value of the **UseOracleConnectionPool** is set to True, and an appropriate value is mentioned in the **IncrPoolSize** binding property to control the number of connections that can be opened by the adapter clients.

- The adapter clients can specify a post-poll statement, an Oracle PL/SQL block, for the **PostPollStatement** binding property. The statement specified in this binding property is executed after the statement specified in the **PollingStatement** binding property is executed.

The adapter wraps the polling statement and the post-poll statement in a transaction and the transaction timeout value is set as the value specified for the **PollingInterval** binding property. Therefore, it is critical to specify a timeout value that is greater than or equal to the time required to process the incoming message and send a reply. If the time taken by the client program to consume the message or execute the post-poll query is more than the timeout value, the transaction is rolled back. If the time taken is less than the timeout value, the adapter commits the transaction and "sleeps" for the remaining time in the poll before performing the next poll.

The adapter suppresses any empty polling responses coming from the Oracle database.

Differences between Polling and Notification

Though polling and notification are both inbound operations, and inform the adapter clients about the data changes in the Oracle database, the following table illustrates some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Polling	Notification
<p>Polling is supported for all the Oracle database versions that are supported by the Oracle Database adapter.</p>	<p>Notification is supported only for Oracle database versions 10.2 and later.</p>
<p>You can either configure the polling interval to check the data available for polling at regular intervals or instantaneously as and when the data is available.</p> <p> Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last change notification.</p>	<p>The data-change notification is always instantaneous.</p>
<p>Polling is initiated by the adapter. The adapter executes a SQL statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.</p>	<p>Notification is initiated by the Oracle database. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement. Notification is a feature of the Oracle database.</p>
<p>You can use the polling statement to read or update data in the Oracle database.</p>	<p>You can use the notification statement to only read data in an Oracle database.</p>
<p>Polling informs you about the actual data that has changed.</p>	<p>Notification informs only about the type of change in the data such as Insert, Update, and Delete.</p>

For more information about:

- How the adapter supports receiving polling-based messages from Oracle database, see [Receiving Polling-based Data-changed Messages](#).

- Receiving polling-based messages from Oracle database using BizTalk Server, see [Polling Oracle Database by Using BizTalk Server](#).
- Receiving polling-based messages from Oracle database using WCF service model, see [Receiving Polling-based Data-changed Messages by Using the WCF Service Model](#).
- Receiving polling-based messages from Oracle database using WCF channel model, see [Receiving Polling-based Data-changed Messages by Using the WCF Channel Model](#).
- Message structure and SOAP actions for performing a polling query, see [Message Schemas for the Polling Operations](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Receiving Database Change Notifications

The Microsoft BizTalk Adapter for Oracle Database supports the ODP.NET Database Change Notification feature. Using this feature, the adapter clients can register a SELECT statement as the notification query on the database, and the database sends a notification to the adapter client as and when the result set of the SELECT statement changes. The database change notification is implemented in the adapter using the OracleDependency class. For more information about the Database Change Support feature in ODP.NET and the OracleDependency class, see <http://go.microsoft.com/fwlink/?LinkId=124801>.

The Oracle Database adapter exposes an inbound operation, Notification, to support database change notification. However, for the database change notification to work with Oracle Database adapter, you must ensure the following:

- Connect to the Oracle database with the underlying Oracle database version 10.2 or later. Oracle database versions prior to 10.2 do not support notifications.
- Connect to Oracle database as a user that has the CHANGE NOTIFICATION privilege to create a notification registration. To grant the CHANGE NOTIFICATION privilege to a user, connect to the Oracle database as a user with administrative privileges, and run the following command at the SQL prompt:

```
grant change notification to <user name>
```

- Decide on a TCP port that can be used by ODP.NET to receive database change notifications from Oracle database. Add the TCP port to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkID=196959>. You must provide the same TCP port number for the **NotificationPort** binding property. For more information about the binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

A typical database change notification using the Oracle Database adapter involves the following:

1. The adapter clients must specify **Notification** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is Polling.
2. The adapter clients must specify a SQL SELECT statement to register for database change notifications in the **NotificationStatement** binding property. The adapter client gets a notification from Oracle database as and when the result set for the specified SQL statement changes.
3. The adapter clients must specify whether the adapter sends a notification to the adapter clients as soon as the listener is started in the **NotifyOnListenerStart** binding property.
4. The notification is sent to the adapter clients as and when the result set of the SELECT statement specified in the **NotificationStatement** binding property is changed.

⚠ Caution

If there is a network outage between the Oracle database and the adapter client, the notifications will not be sent to the adapter clients for the changes done on the Oracle database during the period of network outage, and thereafter. Therefore, you must use the Polling operation instead of the Notification operation for critical scenarios.

Differences between Notification and Polling

Though notification and polling are both inbound operations, and inform the adapter clients about the data changes in the Oracle database, the following table illustrates some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Notification	Polling
Notification is supported only for Oracle database versions 10.2 and later.	Polling is supported for all the Oracle database versions that are supported by the Oracle Database adapter.

<p>The data-change notification is always instantaneous.</p>	<p>You can either configure the polling interval to check the data available for polling at regular intervals or instantaneously as and when the data is available.</p> <p>Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last change notification.</p>
<p>Notification is initiated by the Oracle database. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement. Notification is a feature of the Oracle database.</p>	<p>Polling is initiated by the adapter. The adapter executes a SQL statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.</p>
<p>You can use the notification statement to only read data in an Oracle database.</p>	<p>You can use the polling statement to read or update data in the Oracle database.</p>
<p>Notification informs only about the type of change in the data such as Insert, Update, and Delete.</p>	<p>Polling informs you about the actual data that has changed.</p>

For more information about:

- The binding properties related to the Notification operation, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
- How to use the Notification operation in Oracle Database adapter, see [Receiving Database Change Notifications Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Support for Oracle User-Defined Types

The Microsoft BizTalk Adapter for Oracle Database supports performing operations on artifacts in the Oracle database that contain Oracle User-Defined Types (UDTs). The UDTs can be present in the following artifacts:

- Tables and views containing UDT columns.
- Packages, stored procedures, and functions containing UDT parameters.

What is an Oracle UDT?

Oracle UDTs help in representing complex entities as a "single" object that can be shared among the applications. For example, it is possible to model real-world entities such as "Customers" or "Sales Orders" as objects in the Oracle database. Oracle UDTs are defined in the Oracle database, and they are of the following two types:

- Object types. For example, Oracle Object.
- Collection types. For example, nested table types or VARRAY.

The name of the Oracle UDT is case sensitive, and must be specified in the following way: [SCHEMA_NAME].[UDT_NAME].

How Does the Adapter Support Oracle UDT?

ODP.NET supports UDTs by representing Oracle UDTs defined in the Oracle database as .NET types (custom types). Custom types define the mapping between the Oracle UDT attributes or elements to the .NET members. Custom types can be .NET classes or structures, and can represent either Oracle Objects or Oracle Collections. Owing to the fact that the Oracle Database adapter uses ODP.NET to connect to the Oracle database, it inherits support for Oracle UDTs.

The Oracle Database adapter uses the ODP.NET to specify a custom type mapping to map a .NET custom type to an Oracle UDT in the database. To specify a custom type mapping, the Oracle Database adapter uses a custom type factory. Therefore, in order to use an Oracle UDT, an assembly (.dll file) is required that defines the custom type factory. The Oracle Database adapter enables you to generate an assembly for the custom type factory while generating the metadata for an artifact/operation that contains an Oracle UDT.

Note

The adapter generates the assembly for the Oracle UDTs based on the classes used by the ODP.NET to support Oracle UDTs. For detailed information about how Oracle UDTs are supported in ODP.NET, see <http://go.microsoft.com/fwlink/?LinkId=140697>.

To generate the assembly file for using the Oracle UDTs at design time and then use it later at the run time, the Oracle Database adapter exposes the following binding properties:

- **GeneratedUserTypesAssemblyFilePath** (design time)
- **GeneratedUserTypesAssemblyKeyFilePath** (design time)
- **UserAssembliesLoadPath** (run time)

For information about these binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Performing Operations On Artifacts Containing Oracle UDTs

To perform operations on artifacts containing UDTs using the Oracle Database adapter, you must do the following during design time and run time.

Design Time

You must perform these steps while generation schema for the operation in Visual Studio.

1. Connect to the Oracle database using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the

Add Adapter Service Reference Plug-in. For information about doing so, see [Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in](#).

2. While connecting, in the **Binding Properties** tab of the **Configure Adapter** dialog box, specify appropriate values for the **GeneratedUserTypesAssemblyFilePath** and **GeneratedUserTypesAssemblyKeyFilePath** binding properties. For information about these binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
3. When you are connected to the Oracle database in Visual Studio, browse to the required artifact that contains an Oracle UDT. For information about browsing artifacts, see [Browsing, Searching, and Retrieving Metadata for Oracle Operations](#).
4. Select the required artifact, and then click **OK**. The Oracle Database adapter generates the metadata for the selected operation along with the assembly (.dll file) for the Oracle UDT in the selected artifact. The assembly is created at the location that you specified in the **GeneratedUserTypesAssemblyFilePath** binding property.
5. Proceed with the rest of the steps for building and deploying your project.

Run Time

You must perform these steps in the adapter clients to perform operations on the Oracle UDTs.

In BizTalk Server

- Manually add the Oracle UDT assembly created in step 4 in “Design Time” to the Global Assembly Cache (GAC) on your computer. Alternatively, you can manually copy the Oracle UDT assembly under the BizTalk Server installation location. For BizTalk Server 2010, typically this is <installation drive>:\Program Files\Microsoft BizTalk Server 2010.
- While configuring the BizTalk Server WCF-Custom or WCF-OracleDB port, in the **Binding** tab, specify the location of the Oracle UDT assembly for the **UserAssembliesLoadPath** binding property. For information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

In Visual Studio

- Manually add the Oracle UDT assembly created in step 4 in “Design Time” to the Global Assembly Cache (GAC) on your computer. Alternatively, you can manually copy the Oracle UDT assembly to the same location as the project executable file, which typically is under the project’s \bin\Debug folder.
- Specify the location of the Oracle UDT assembly for the **UserAssembliesLoadPath** binding property. For information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

How does the Adapter Handle Transactions?

Microsoft BizTalk Adapter for Oracle Database does not initiate a transaction while performing an operation on the Oracle database. Instead, the adapter performs the operations using the transaction context provided by the adapter clients. In order to perform operations in a transaction using the Oracle Database adapter, you must:

- Enable transactions in the adapter clients. For example, to enable transactions in BizTalk Server 2010, you must select the **Use Transaction** check box in the **Transactions** area of the **Messages** tab for a WCF-Custom or WCF-OracleDB port.
- Set the value of the **UseAmbientTransaction** binding property to **True** in the adapter. For more information about the binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

◆ Important

To use the adapter to perform transactions on the Oracle database, you must have installed the **Oracle Services For Microsoft Transaction Server** component, while installing the Oracle client, on the computer running the adapter client.

Transactions in the Outbound Operations

The Oracle Database adapter performs an outbound operation in a single transaction. For composite operations, all the operations are performed in a single transaction but using different ODP.NET connections. For more information about the outbound operations surfaced by the Oracle Database adapter, see [How Does the Adapter Surface Oracle Metadata?](#).

Transactions in the Inbound Operations

The Oracle Database adapter exposes the following two inbound operations:

- **Polling:** The polling statement and the post-poll statement (if specified) are executed in a transaction, whereas, the polled data available statement is executed in a different transaction. Similarly, the polling statement and the post-poll statement are executed using the same ODP.NET connection, whereas, the polled data available statement is executed using a different ODP.NET connection.
- **Notification:** The notification operation is performed in a transaction using a single ODP.NET connection.

For more information about the inbound operations surfaced by the Oracle Database adapter, see [How Does the Adapter Surface Oracle Metadata?](#).

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

Streaming Support for LOB Data Types

The Oracle database supports streaming on large object (LOB) data types. The Oracle Database adapter supports message streaming, which makes it possible to stream LOB data end-to-end between the Oracle database and an adapter client. However, streaming is not supported in the same manner across all programming models when you use the adapter.

The following shows how end-to-end streaming of LOB data types is supported by the adapter across different programming models.

Operation	WCF Channel Model	WCF Service Model	BizTalk Server
Table/View Insert operation	Not supported	Not supported	Not supported
Table/View Select operation	Supported	Not supported	Supported
Table/View Update operation	Not supported	Not supported	Not supported
Table/View Delete operation	Not supported	Not supported	Not supported
Table/View ReadLOB operation	Supported; however, the ReadLOB operations is surfaced primarily to support streaming in the WCF service model, it is not recommended for use in the WCF channel model. Use a Select operation or the SQLEXECUTE operation instead.	Supported	The ReadLOB operation is not supported for BizTalk Server. Use a Select operation instead.
Table/View UpdateLOB operation	Supported	Not supported	Supported
SQLEXECUTE operation	Supported in the response	Not supported	Supported in the response
Stored procedure and function operation	Supported in the response	Not supported	Supported in the response
POLLINGSMTP operation	Supported	Not supported	Supported

For more comprehensive information about how streaming of LOB data types is supported by the adapter and how it is supported when you use various programming models with the adapter, see [Streaming and the Oracle Database Adapter](#).

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

Other Features Supported by the Adapter

In addition to the features discussed throughout the topics of [Overview of BizTalk Adapter for Oracle Database](#), the Oracle Database adapter also provides the following features that are useful for adapter clients:

- **Support for configuring adapters using binding properties.** Adapter clients can configure the Oracle Database adapter by specifying certain binding properties. For example, clients can configure the adapter to use the ODP.NET connection pool by setting the **UseOracleConnectionPool** binding property. For more information, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
- **Support for null values for operation parameters.** Adapter clients can provide null values for operation parameters using the "nil" attribute in the input XML.
- **Support for dynamic ports in BizTalk.** Through the BizTalk WCF-Custom adapter, the Oracle Database adapter supports a dynamic port that enables dynamic routing of messages from BizTalk Server based on the message context properties. For more information, see [Configuring Dynamic Ports](#).
- **Support for performance counters.** The Oracle Database adapter supports WCF-based performance counters for use by adapter clients. For more information about performance counters, see [Using Performance Counters](#).

See Also

Concepts

[Overview of BizTalk Adapter for Oracle Database](#)

New and Depreciated Features in BizTalk Adapter for Oracle Database

This section lists the new and deprecated features in Microsoft BizTalk Adapter for Oracle Database.

New Features in the Oracle Adapter

The following are the new features introduced in this release of Oracle Database adapter.

Technology-Related Features

Feature	Comment
New way of connecting to the Oracle database	Apart from connecting to the Oracle database using the net service name in the tnsnames.ora file (as in the previous version of the adapter), the adapter clients can now also connect to the Oracle database directly by specifying the connection parameters, and thus eliminating the need to use a net service name or the tnsnames.ora file. Not requiring the tnsnames.ora file to connect to the Oracle database saves you from the hassle of manually updating the connection parameters (net service name) in the tnsnames.ora file on every client computer when you add or update Oracle servers in your environment. For more information, see Establishing a Connection to the Oracle Database .
Support for Windows Authentication	The adapter clients can use Windows Authentication to connect to the Oracle database. Windows Authentication enables you to determine the user's identity based on the Windows logon credentials, and thus helps you to leverage the built-in security of the Windows environment. For more information about Windows Authentication in the Oracle Database adapter, see Connecting to the Oracle Database Using Windows Authentication .

Operations-Related Features

Feature	Comment
Support for specifying inline values in the Insert operation	You can use the InlineValue attribute in the Insert operation to insert computed values into tables or views in the Oracle database. This is an optional attribute and is available for all simple data records in a multiple record Insert operation. If you specify a value for this attribute, it overrides the specified value of a record. For more information about the InlineValue attribute, see Performing Basic Insert, Update, Delete, and Select Operations on Oracle Tables and Views .
Enhanced polling	The Oracle Database adapter now supports receiving "polling-based" data-changed messages by using stored procedures, functions, or packaged procedures or functions to periodically poll the Oracle database. In addition to the SELECT statement, now you can specify a stored procedure, function, or packaged procedure or function as a polling statement that the adapter executes periodically to poll the Oracle database. For more information about polling, see Support for Receiving Polling-based Data-changed Messages .
Support for Oracle User-Defined Types (UDTs)	The Oracle Database adapter supports performing operations on artifacts in the Oracle database that contain Oracle UDTs. For information about UDT support, see Support for Oracle User-Defined Types .
Support for composite operations	The Oracle Database adapter enables adapter clients to perform composite operations on the Oracle database. A composite operation can include any number of the following operations, and in any order: <ul style="list-style-type: none"> Operations on tables and views. Stored procedures, functions, and procedures or functions within packages that are surfaced as operations in the adapter. <p>For more information about composite operations, see Performing Composite Operations.</p>

Support for executing stored procedures in the schemas not owned by the user	The Oracle Database adapter allows you to execute stored procedures in a schema even if the current user is not the owner of the schema, provided the user has permissions on the schema in Oracle. However, if the stored procedure uses RECORD types, they must be defined in the same schema as the stored procedure. For information about executing stored procedures using the Oracle Database adapter, see Operations on Functions and Stored Procedures .
Support for database change notifications	Adapter clients can receive database change notifications from Oracle database based on a triggering SELECT statement. The notification is sent by Oracle database to the adapter clients as and when the result set for the SELECT statement changes. For more information about database change notifications, see Receiving Database Change Notifications .
Support for synonyms	Adapter clients can perform operations on synonyms created for tables, views, stored procedures, functions and packages. For information about synonyms, and how you can use the Oracle Database adapter to perform operations on synonyms, see Operations on Synonyms .
Support for boolean parameters and PL/SQL table types	Adapter clients can perform operations in stored procedures and functions that contain boolean parameters and PL/SQL table type.

Other Features

Feature	Comment
New way of using the adapter in BizTalk Server	The Oracle Database adapter can be used in BizTalk either as a WCF-Custom port or a WCF-OracleDB port. If you want to use the Oracle Database adapter through a WCF-Custom port, you do not need to add the WCF-Custom port to the BizTalk Server Administration console because the WCF-Custom port is added to the BizTalk Server Administration console by default. However, if you want to use the Oracle Database adapter through a WCF-OracleDB port, you must first add the WCF-OracleDB adapter to the BizTalk Server Administration console. For more information, see Adding the Oracle Database Adapter to BizTalk Server Administration Console .

Deprecated Features in the Oracle Adapter

The following table lists the feature that are deprecated in the current version of the Oracle Database adapter.

Feature	Comment
Binding properties	The PollingRetryCount , TransactionIsolationLevel , and LongDataTypeColumnSize binding properties are deprecated in the current version of the Oracle Database adapter.  Note To set the transaction isolation level for the inbound operations, you must set appropriate value by adding the service behavior while configuring the receive port. For instructions on how to set the transaction isolation level, see Configure Transaction Isolation Level and Transaction Timeout .

Changes Since the Last Release

This section lists the changes that have happened in the Oracle Database adapter since the last release as a result of bug fixes or code changes.

General

- In the previous version of the Oracle Database adapter, for parameters of type IN OUT REF CURSOR:
 - If there was no change to the REF CURSOR value inside the stored procedure, the value of the output REF CURSOR use to be NULL instead of the data passed in.
 - It allowed input and output data to be in different formats.

This has changed in the current release of the Oracle Database adapter, and now:

- If there was no change to the REF CURSOR value inside the stored procedure, the value of the output is the same as the value in the input REF CURSOR.
- The input and output data in the REF CURSOR is of the same type.
- Incorrect behavior of the "nil" attribute: For all simple data types, if you set the value of the nil attribute to "true", and a value for the field or parameter is present then the Oracle database adapter incorrectly passes the specified value instead of NULL. As a workaround, if you want to pass NULL value for a field or parameter, you must ensure that no value for the field or parameter is specified. For example, to pass NULL value for a field called "name":

```
<name xsi:nil="true"/>
```

- In the current version of the Oracle Database adapter, the exceptions thrown for various actions are not same as the previous version of the adapter.
- Unlike in the previous version of the adapter, for the Real, Float, and Long data types, extra zeroes (0) at the end of the value in the result set of the Select operation are not truncated in the current version of the Oracle Database adapter. Moreover, in the current version of the adapter, the result set of the Select operation always returns a value with precision 8 for the Real, Float, and Long data types.
- Handling of data for the RECORD types: In the earlier version of the adapter, for parameters of RECORD types in a stored procedure, the adapter always passed NULL values for nodes that were either not present or marked as 'nil' in the request XML. In the current version of the adapter, the value passed for these nodes depend on the value of the **SkipNilNodes** binding property. For more information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
- Outbound operations: In the previous version of the Oracle Database adapter, if you executed stored procedures in which some parameters are not specified in the input XML file, a NULL value was sent for those parameters. Even if default values were specified for these parameters in the stored procedure, still a NULL value was sent for those parameters. In the current version of the adapter, no value is sent for the parameters that do not have a value specified in the input XML file. If a default value is specified in the stored procedure, the Oracle database uses that value because no value was sent by the adapter. If a NULL value needs to be sent, the user needs to specify a NULL node in the input XML file by setting the value of "nil" attribute to "true."
- The Oracle Database adapter now supports command timeout.
- The UpdateLOB operation must be performed as part of a transaction. To ensure this, the value of the **UseAmbientTransaction** binding property must be set to **True**.

BizTalk Scenario

- Outbound operations: In the previous version of the Oracle Database adapter, the outbound operations were executed within a non-distributed transaction, and this transaction was disjoint from the BizTalk MessageBox transaction. In the current version of the adapter, if the **UseAmbientTransaction** binding property is "True," the operations on the Oracle database and on the BizTalk MessageBox database are performed within the same distributed transaction. For more information about transactions in the Oracle Database adapter, see [How does the Adapter Handle Transactions?](#)
- Inbound operations: You cannot use a Request-Response receive port in BizTalk Server for the inbound operations using the Oracle Database adapter. Only One-Way receive ports can be used.

Other Scenarios

- Outbound operations: In the previous version of the Oracle Database adapter, the operations were performed transactionally. For example, if there were 25 rows to be inserted in an INSERT operation, the operation would be performed in the same local transaction. In the current version of the adapter, the adapter no longer initiates a

transaction. If the user wants these rows to be inserted within the same transaction, it is the user's responsibility to execute the operation within a System.Transactions Transaction Scope. The user also needs to set the value of the **UseAmbientTransaction** property to **True**. For more information about transactions in the Oracle Database adapter, see [How does the Adapter Handle Transactions?](#).

- Outbound operations: In the current version of the Oracle Database adapter, all operations performed on the same IRequestChannel/proxy object might not be performed on the same physical connection to the Oracle Database.
- WCF Channel Model: The Oracle Database adapter no longer supports IReplyChannel while using the WCF Channel model. However, you can still use IInputChannel for performing inbound operations. Further, with regards to transactions, the adapter relies on the WCF Dispatcher initiated transaction to execute the polling statement and post poll statement against the Oracle database. The transaction isolation level and the time out of the WCF Dispatcher initiated transaction can be controlled by setting appropriate values in the ServiceBehavior.

See Also

Other Resources

[Understanding BizTalk Adapter for Oracle Database](#)

Limitations of BizTalk Adapter for Oracle Database

General

The following are known limitations for the Microsoft BizTalk Adapter for Oracle Database:

- Barring some exceptions, the Oracle Database adapter is compatible with the previous release of the adapters. For a list of changes that has happened since the last release, see [New and Deprecated Features in BizTalk Adapter for Oracle Database](#).

Note

You can modify the BizTalk projects for previous version of the Oracle database adapter to use the new WCF-based Oracle Database adapter. For more information, see [Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#).

- The Oracle Database adapter does not support XML Types.
- The SQLEXECUTE operation does not return values for OUT or IN OUT parameters to procedures, functions, or packages. For this reason, you must invoke procedures, functions, and packages by using the dedicated operations that the Oracle Database adapter exposes for these Oracle artifacts.
- When retrieving data from the Oracle database using proxy programming, the Oracle Database adapter does not deserialize XML messages that have more than 65536 nodes. Make sure the response message has nodes less than or equal to 65536. You can work around this limitation by modifying the app.config file for your application. For instructions, see [Troubleshooting Operational Issues](#).
- The Oracle Database adapter takes input strings and constructs SQL commands that are then executed by the adapter. However, the input string might contain other SQL commands that also get executed and might break the operation contract.

Consider a scenario where the adapter provides an input REF CURSOR to a stored procedure. In such a scenario, the adapter client must provide a command that, when executed, obtains the REF CURSOR. The adapter then passes on the REF CURSOR to the stored procedure. However, if the command for obtaining the REF CURSOR performs some additional modifications to the database, the operation contract for executing the stored procedure is broken.

- The Oracle Database adapter supports UDT nesting only up to two levels.
- When using the adapters with BizTalk Server, if the credentials on the WCF-custom send port are incorrect, the request messages are not processed. After you specify the correct credentials, the message is sent to the Oracle database and a response is received. However, the response message is not available to the out port. In such scenarios, you may need to restart the host instance.
- The Oracle Database adapter does not support the BFILE data type inside complex types (such as RECORD type, TABLE type, UDT, and VARRAY).
- The Oracle Database adapter does not support User-Defined Types (UDTs) that have circular references.
- The Oracle Database adapter does not support records that contain fields of type PL/SQL tables of RECORD type.
- The Oracle Database adapter does not enable clients to set the value of the first element in a VARRAY to NULL.
- Except for PL/SQL tables, the Oracle Database adapter does not support UDTs that are defined inside a package.

The following are known limitations for the Oracle Database adapter due to the limitation of ODP.NET:

- For Oracle data types that take decimal values, ODP.NET does not throw an exception if the input value contains alphabetic characters. Because the Oracle Database adapter uses ODP.NET to interface with the Oracle database, the adapter too does not throw an exception when passing alphabetic characters. For example:
 - Passing a value "54r" for an insert operation does not throw an exception; the value "54" is inserted instead.
 - Passing a value "r54" for an insert operation does not throw an exception; the value "0" is inserted instead.
- Because of an ODP.NET limitation, the Oracle Database adapter does not support the use of overloaded procedures using strongly-typed and weakly-typed REF CURSORS. Internally, the adapter treats both the strongly-typed and weakly-typed REF CURSORS as just REF CURSORS.
- The Oracle Database adapter does not support PL/SQL tables that are not indexed by a numeric field.
- The Oracle Database adapter does not support associative arrays that do not contain any element.
- The Oracle Database adapter does not support UDTs that contain the TimeStamp data type with local time zone attributes (TimeStampLTZ).
- The Oracle Database adapter does not support UDTs that contain a "." (period) in their names.
- The Oracle Database adapter does not support UDTs that contain BLOB, CLOB, and NCLOB data types as an IN OUT parameter.
- The Oracle Database adapter does not support Varray of Varray of the following simple types: BFILE, IntervalDS, IntervalYM, TimeStampLTZ, and TimeStampTZ.
- Due to the limitation of associative arrays, PL/SQL tables or PL/SQL tables of records that contain any of the following data types are not supported in the Oracle Database adapter:
 - BFILE
 - BLOB
 - CLOB
 - IntervalDS
 - IntervalYM
 - Long
 - NCLOB
 - RowID
 - TimeStamp
 - TimeStampLTZ
 - TimeStampTZ

See Also

Other Resources

Oracle Database Adapter Migration Tutorial

The Oracle Database adapter migration tutorial contains detailed instructions on how to migrate BizTalk projects created using the BizTalk ODBC Adapter for Oracle Database (shipped with BizTalk Server) to use the WCF-based Microsoft BizTalk Adapter for Oracle Database (shipped with BizTalk Adapter Pack 2010).

Also, the topics in the [Developing BizTalk Applications](#) section provide detailed instructions on how to use the Oracle Database adapter with BizTalk Server.

In This Section

- [Tutorial: Migrating BizTalk Projects](#)

See Also

Other Resources

[Getting Started](#)

Tutorial: Migrating BizTalk Projects

The BizTalk ODBC Adapter for Oracle Database that shipped with Microsoft BizTalk Server differs from the WCF-based Oracle Database adapter in many aspects, including:

- The design-time experience of creating a BizTalk project.
- The metadata retrieval experience.
- Schema file name and namespace.
- Data type mappings.
- The operations that can be performed using the adapter.
- Physical port configuration in the BizTalk Server Administration console

These differences are explained in the topics within [Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#).

However, you can make changes to the BizTalk project that was created using the BizTalk ODBC Adapter for Oracle Database and make it work with the WCF-based Oracle Database adapter.

This tutorial provides instructions on the changes you should make to the existing BizTalk project created using the BizTalk ODBC Adapter for Oracle Database.

Note

In this tutorial, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database will be referred to as "vPrev Oracle Database adapter." Similarly, a BizTalk project that uses the vPrev Oracle Database adapter will be referred to as "vPrev BizTalk project."

Sample Used for the Tutorial

This tutorial is based upon a sample (Oracle_Migration) that demonstrates how to migrate a vPrev BizTalk project. The sample is provided with Microsoft BizTalk Adapter Pack. For more information, see [Samples](#).

Prerequisites

- You must have a vPrev BizTalk project. This tutorial involves a BizTalk project that performs an Insert operation on a CUSTOMER table. The CUSTOMER table is created under the SCOTT schema by running the SQL scripts provided with the BizTalk Adapter Pack samples.
- You must have a request message to perform an Insert operation on the Oracle database using the vPrev Oracle Database adapter. The request message must conform to the schema of the Insert operation generated using the vPrev Oracle Database adapter.
- You must have completed the steps in [Before You Develop BizTalk Applications](#).

Understanding a BizTalk Project Created Using the Previous Version of the Adapter

The key constituents of a vPrev BizTalk project created are:

- **BizTalk orchestration.** This is a simple orchestration that picks request messages from a file location, sends the request message to the Oracle database using an Oracle send-receive port, receives the response, and saves it to another file location.
- **Schema for the operation you wish to perform on the Oracle database.** This tutorial involves a BizTalk project that performs an Insert operation on the CUSTOMER table in the SCOTT schema. The CUSTOMER table is created under the

SCOTT schema by running the SQL scripts provided with the BizTalk Adapter Pack samples. The schema generated for the CUSTOMER table is CUSTOMERService_CUSTOMER_x5d.xsd. This schema is generated using the vPrev Oracle Database adapter.

 **Note**

Unlike the WCF-based Oracle Database adapter, the vPrev Oracle Database adapter does not support generating meta data for specific operations on an Oracle database table. By default, the adapter generates schema for all the operations supported on the table. For more such differences between the vPrev Oracle Database adapter and the WCF-based Oracle Database adapter, see [Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#).

- **Request message.** The request message to perform an Insert operation on the CUSTOMER table. The schema of the request message conforms to the schema of the Insert operation as surfaced by the previous version of the Oracle Database adapter.

How to Migrate a BizTalk Project Created Using the Previous Version of the Adapter

The goal of this migration tutorial is to enable you to send a request message, which conforms to schema generated by vPrev Oracle Database adapter, using a WCF-Custom port that can only process messages conforming to the WCF-based Oracle Database adapter. So, in short, the migration exercise involves configuring the WCF-Custom port to process messages that do not conform to the WCF-based Oracle Database adapter's schema.

However, to be able to configure the WCF-Custom port appropriately, you must perform the following tasks:

- Generate metadata for the Insert operation on the SCOTT.CUSTOMER table using the WCF-based Oracle Database adapter.
- Map the request message for performing an Insert operation using the vPrev Oracle Database adapter to a request message for performing an Insert operation using the WCF-based Oracle Database adapter.
- Map the response message received using the WCF-based Oracle Database adapter to the response message for the vPrev Oracle Database adapter.
- Create a WCF-Custom Oracle send-receive port in the BizTalk Server Administration console.
- Configure the WCF-Custom port to use the request and response mappings.

In This Section

- [Step 1: Modify the vPrev BizTalk Project](#)
- [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#)
- [Step 3: Test the Migrated Application](#)

See Also

Other Resources

[Oracle Database Adapter Migration Tutorial](#)

Step 1: Modify the vPrev BizTalk Project



Step 1

Time to complete: 10 minutes

Objective: In this step, you make the following changes to the existing vPrev BizTalk project:

- Generate metadata for the Insert operation on the SCOTT.CUSTOMER table using the WCF-based Oracle Database adapter.
- Map the request message for performing an Insert operation using the vPrev Oracle database adapter to a request message for performing an Insert operation using the WCF-based Oracle Database adapter.
- Map the response message received using the WCF-based Oracle Database adapter to the response message for the vPrev Oracle database adapter.

Prerequisites

- You must have a vPrev BizTalk project to perform an Insert operation on the SCOTT.CUSTOMER table in the Oracle database.

To modify the vPrev BizTalk project

1. Generate metadata for the Insert operation on the SCOTT.CUSTOMER table using the WCF-based Oracle Database adapter. You can use the Consume Adapter Service BizTalk Project Add-in to generate metadata.

For instructions on how to generate metadata, see [Retrieving Metadata for Oracle Operations in Visual Studio](#). After the schema is generated, a file with the name similar to *OracleDBBindingSchema.xsd* is added to the BizTalk project. This file contains the schema for sending a message to perform an Insert operation on the SCOTT.CUSTOMER table in the Oracle database using the WCF-based Oracle Database adapter.

2. Generating the metadata for the Insert operation also creates a port binding file. In the next step, this binding file will be used to create a WCF-Custom send port to send messages to the Oracle database. The SOAP action for the operation is also set to the operation for which you generated metadata. For example, if you generate metadata for the Insert operation, the operation name in the SOAP action on the send port will be "Insert". However, the operation name on the logical send port that you create as part of the orchestration could be different, for example, "Operation_1". As a result, when you send messages to the Oracle database using the send port, you get an error. To prevent this, make sure the operation name on the logical send port in your orchestration is the same as the operation name for which you generated metadata.

So, in case of this tutorial, because you generate metadata for the Insert operation, change the name of the logical send port operation to "Insert".

3. For the request message, map the schema generated using vPrev Oracle database adapter to the schema generated using the WCF-based Oracle Database adapter.

- a. Add a BizTalk mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.

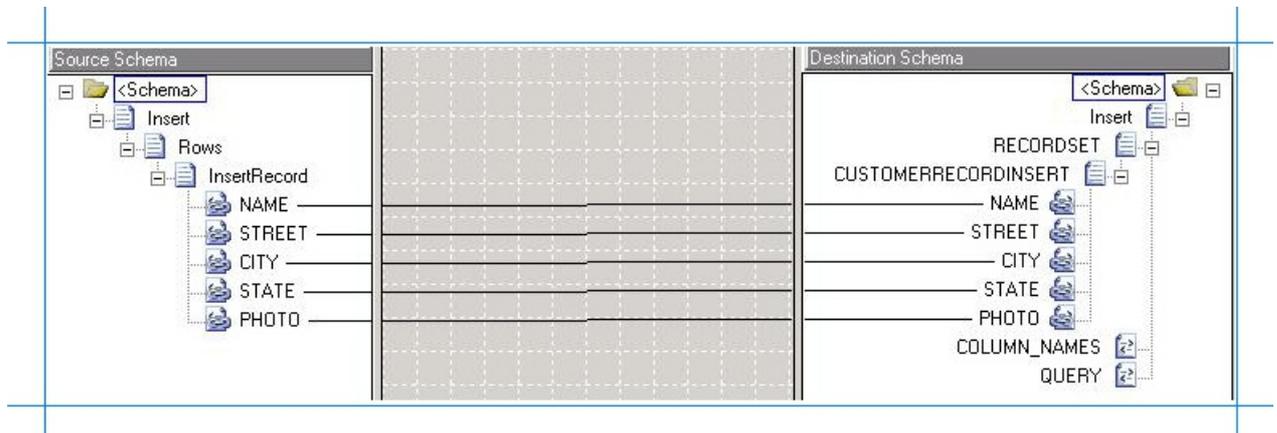
In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **RequestMap.btm**. Click **Add**.

- b. From the Source Schema pane, click **Open Source Schema**.

- c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the request message for the vPrev Oracle database adapter. For this tutorial, select *Oracle_Migration.CUSTOMERService_CUSTOMER_x5d*. Click **OK**.

- d. In the **Root Node for Source Schema** dialog box, select *Insert* and click **OK**.

- e. From the Destination Schema pane, click **Open Destination Schema**.
- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the request message for the WCF-based Oracle Database adapter. For this tutorial, select *Oracle_Migration.OracleDBBindingSchema*. Click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *Insert* and click **OK**.
- h. Map the respective elements in both the schemas as illustrated in the following figure.



- i. Save the map.

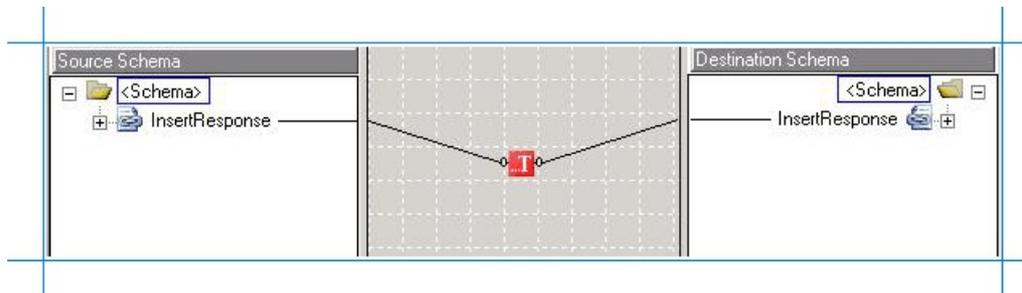
4. For the response message, map the schema generated using vPrev Oracle database adapter to the schema generated using the WCF-based Oracle Database adapter.

- a. Add a BizTalk mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.

In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **ResponseMap.btm**. Click **Add**.

- b. From the Source Schema pane, click **Open Source Schema**.
- c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the response message for the WCF-based Oracle Database adapter. For this tutorial, select *Oracle_Migration.OracleDBBindingSchema*. Click **OK**.
- d. In the **Root Node for Source Schema** dialog box, select *InsertResponse* and click **OK**.
- e. From the Destination Schema pane, click **Open Destination Schema**.
- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the response message for the vPrev Oracle database adapter. For this tutorial, select *Oracle_Migration.CUSTOMERService_CUSTOMER_x5d*. Click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *InsertResponse*, and then click **OK**.
- h. You will notice that the schema for the response message conforming to the WCF-based Oracle Database adapter contains an additional *InsertResult* element. You must remove this from the schema and map the *InsertResponse* element in both the schemas.

To do so, from the **Toolbox**, drag the **String Left Trim** functoid and drop it on the mapper grid. Connect the **InsertResponse** element in the source schema to the functoid. Similarly, connect the **InsertResponse** element in the destination schema to the functoid. The following figure illustrates how the two elements are mapped via the functoid.



Note

For more information about the String Left Trim functoid, see "String Left Trim Functoid" at <http://go.microsoft.com/fwlink/?LinkId=105774>.

i. Save the map.

5. Save and build the BizTalk solution. Right-click the solution, and then click **Build Solution**.

6. Deploy the solution. Right-click the solution, and then click **Deploy Solution**.

Next Steps

Create a WCF-custom send port and configure it to use the maps you created in this step, as described in [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#).

See Also

Concepts

[Tutorial: Migrating BizTalk Projects](#)

Step 2: Configure the Orchestration in BizTalk Server Administration Console



Time to complete: 10 minutes

Objective: In this step, you perform the following tasks:

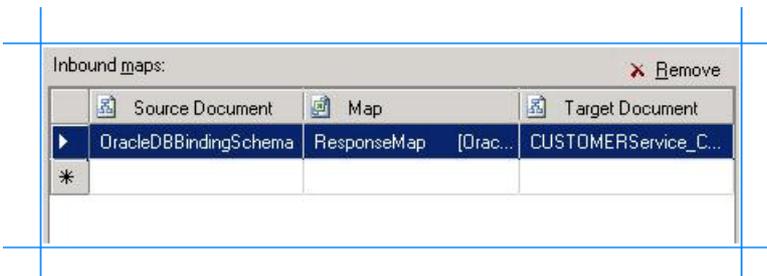
- Create a WCF-Custom send-receive port to send and receive messages from the Oracle database using the Oracle Database adapter. Configure this port to use the maps you created in the previous step.
- Configure the orchestration you deployed in the last step to use the WCF-Custom port.

Prerequisite

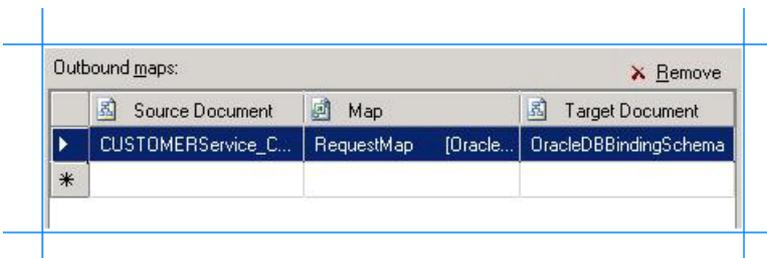
- You must have deployed the BizTalk orchestration for which you want to configure the WCF-Custom port.

To create a WCF-Custom port

1. When you generate schema for an operation on the Oracle database using Consume Adapter Service BizTalk Project Add-in, a binding file is also added to the BizTalk project. You can import this binding file into your BizTalk application to create a WCF-Custom send-receive port. For instructions on importing a binding file, see [Importing Bindings](#).
2. After you import the binding file, a send port is created under the **Send Ports** folder in the BizTalk Server Administration console.
3. Right-click the WCF-Custom port and click **Properties**.
4. From the left pane of the send port properties dialog box, click the **General** tab. From the right pane, click **Configure**.
5. In the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, and specify the credentials to connect to an Oracle database.
6. Click **OK**.
7. From the left pane of the send port properties dialog box, click **Inbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **ResponseMap**.



8. From the left pane of the send port properties dialog box, click **Outbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **RequestMap**.



9. Click **OK**.

To configure the BizTalk application

1. In the BizTalk Server Administration console, expand **BizTalk Group**, expand **Applications**, and expand the BizTalk Application where the orchestration is deployed.
2. Right-click the BizTalk application, and then select **Configure**.
3. From the left pane, click the orchestration to configure. From the right pane, from the **Host** drop-down list, select a BizTalk host instance.
4. Under the **Bindings** box, map the logical ports of the BizTalk orchestration to the physical ports in the BizTalk Server Administration console.
 - a. Select the file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
 - b. Select the file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
 - c. Select the WCF-Custom send port you created earlier in this topic.
 - d. Click **OK**.

For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Next Steps

You have now completed migration of your vPrev BizTalk project to a BizTalk project that sends messages to the Oracle database using the WCF-based Oracle Database adapter. You must now test the migrated BizTalk application by sending a request message to perform an Insert operation on the Oracle database, as described in [Step 3: Test the Migrated Application](#).

See Also

Concepts

[Tutorial: Migrating BizTalk Projects](#)

Step 3: Test the Migrated Application



Time to complete: 5 minutes

Objective: In this step, you will test the migrated application by performing an Insert operation on the SCOTT.CUSTOMER table. To do this, you drop a request message that conforms to the schema generated using the vPrev Oracle Database adapter.

Prerequisites

- Configure the BizTalk application by mapping the logical ports in the BizTalk orchestration to physical ports in the BizTalk Server Administration console.
- Configure the BizTalk application to use the WCF-Custom send port for the WCF-based Oracle Database adapter.

To test the migrated application

1. From the Oracle_Migration folder, copy the OracleInsert.xml request message. This request message conforms to the schema generated by the vPrev Oracle Database adapter. Using the outbound map, the WCF-Custom send port converts this to conform to the schema for the WCF-based Oracle Database adapter and sends it to the Oracle database.

```
<ns0:Insert xmlns:ns0="http://schemas.microsoft.com/[OracleDb://ADAPTER/SCOTT/Tables/CUSTOMER]">
  <ns0:Rows>
    <ns0:InsertRecord>
      <ns0:NAME>Customer_1</ns0:NAME>
      <ns0:STREET>Street_1</ns0:STREET>
      <ns0:CITY>City_1</ns0:CITY>
    </ns0:InsertRecord>
    <ns0:InsertRecord>
      <ns0:NAME>Customer_2</ns0:NAME>
      <ns0:STREET>Street_2</ns0:STREET>
      <ns0:CITY>City_2</ns0:CITY>
    </ns0:InsertRecord>
  </ns0:Rows>
</ns0:Insert>
```

2. Paste the request message to the folder mapped to the file receive location.
3. The orchestration consumes the request message and sends it to the Oracle database. The response from the Oracle database is received in the schema that conforms with the schema of the WCF-based Oracle Database adapter. Using the inbound map, the WCF-Custom send port converts this to the schema for the vPrev Oracle Database adapter. The response from the Oracle database is saved to the other file location defined as part of the orchestration. The response for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<ns0:InsertResponse xmlns:ns0="http://schemas.microsoft.com/[OracleDb://ADAPTER/SCOTT/Tables/CUSTOMER]"></ns0:InsertResponse>
```

See Also

Concepts

[Tutorial: Migrating BizTalk Projects](#)

Community Resources

There are two principal sources of information from Microsoft about the BizTalk Adapter Pack:

- The documentation and samples installed with the BizTalk Adapter Pack.
- The Microsoft BizTalk Adapter Pack forums.

BizTalk Adapter Pack Documentation

You can install the BizTalk Adapter Pack documentation when you install the BizTalk Adapter Pack. (For information about how to install the BizTalk Adapter Pack, see the installation guide. The installation guide is typically available under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.) The core documentation covers the concepts and procedures required to use the BizTalk Adapter Pack.

After you install the BizTalk Adapter Pack Setup program, you can access the documentation through the **Microsoft BizTalk Adapter Pack** program group on the Start menu.

BizTalk Adapter Pack Samples

This BizTalk Adapter Pack release provides samples demonstrating how to use the adapters with:

- BizTalk Server.
- The Windows Communication Foundation (WCF) channel model.
- The WCF service model.

In addition, migration samples are provided.

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>.

BizTalk Adapter Pack Forums

You can post your queries about the BizTalk Adapter Pack on the forums.

Resource	Location
BizTalk Adapter Pack Forums	http://go.microsoft.com/fwlink/?LinkId=87695

Getting Assistance with the BizTalk Adapter Pack

Should you encounter any issues with the BizTalk Adapter Pack, you must mention the product ID (PID) for the BizTalk Adapter Pack in all communications with Microsoft Support. The PID is available in a file, pid.txt, installed by the setup wizard, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

The end-user license agreement (EULA) for the BizTalk Adapter Pack is available in a file, EULA.rtf, installed by the setup wizard, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack.

See Also

Other Resources

[Getting Started](#)

Frequently Asked Questions

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle Database.

What are the supported Oracle server versions?

The Oracle Database adapter can be used to connect to the following versions of the Oracle database:

- Oracle database version 9.2
- Oracle database version 10.1
- Oracle database version 10.2
- Oracle database version 11.1
- Oracle database version 11.2

What are the supported Oracle client versions?

The Oracle Database adapter supports Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7, Oracle Data Access Components for Oracle Client 11.1.0.7.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see "Oracle Data Provider for .NET " at <http://go.microsoft.com/fwlink/?LinkId=197108>.

What are the supported BizTalk versions for the BizTalk Adapter Pack?

The BizTalk Adapter Pack is supported with Microsoft BizTalk Server 2010.

In which user context should the setup wizard be run?

You must always run the BizTalk Adapter Pack setup wizard under a user context with administrative privileges on the computer.

Does the Microsoft BizTalk Adapter Pack support tracing?

Microsoft BizTalk Adapter Pack enables adapter clients to activate Windows Communication Foundation (WCF) tracing and adapter-specific tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

Where are the trace files stored?

The trace files are stored at a location that the adapter clients specify. The adapter clients can specify this location while enabling tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

How do I view the traces?

You can use the WCF Service Trace Viewer tool to view the traces. For more information about the tool, see <http://go.microsoft.com/fwlink/?LinkId=91243>.

See Also

Other Resources

[Getting Started](#)

Planning and Architecture

This section describes the architecture for the Microsoft BizTalk Adapter for Oracle Database. An understanding of the Oracle Database adapter architecture can help you:

- Understand the relationship between the Oracle Database adapter and the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
- Understand security boundaries, so that you can better secure data in your solution.
- Understand the Oracle Database adapter binding properties.
- Troubleshoot installation issues.

The following topics describe the architecture of end-to-end solutions that use the Oracle Database adapter to operate on an Oracle database and the internal architecture of the Oracle Database adapter.

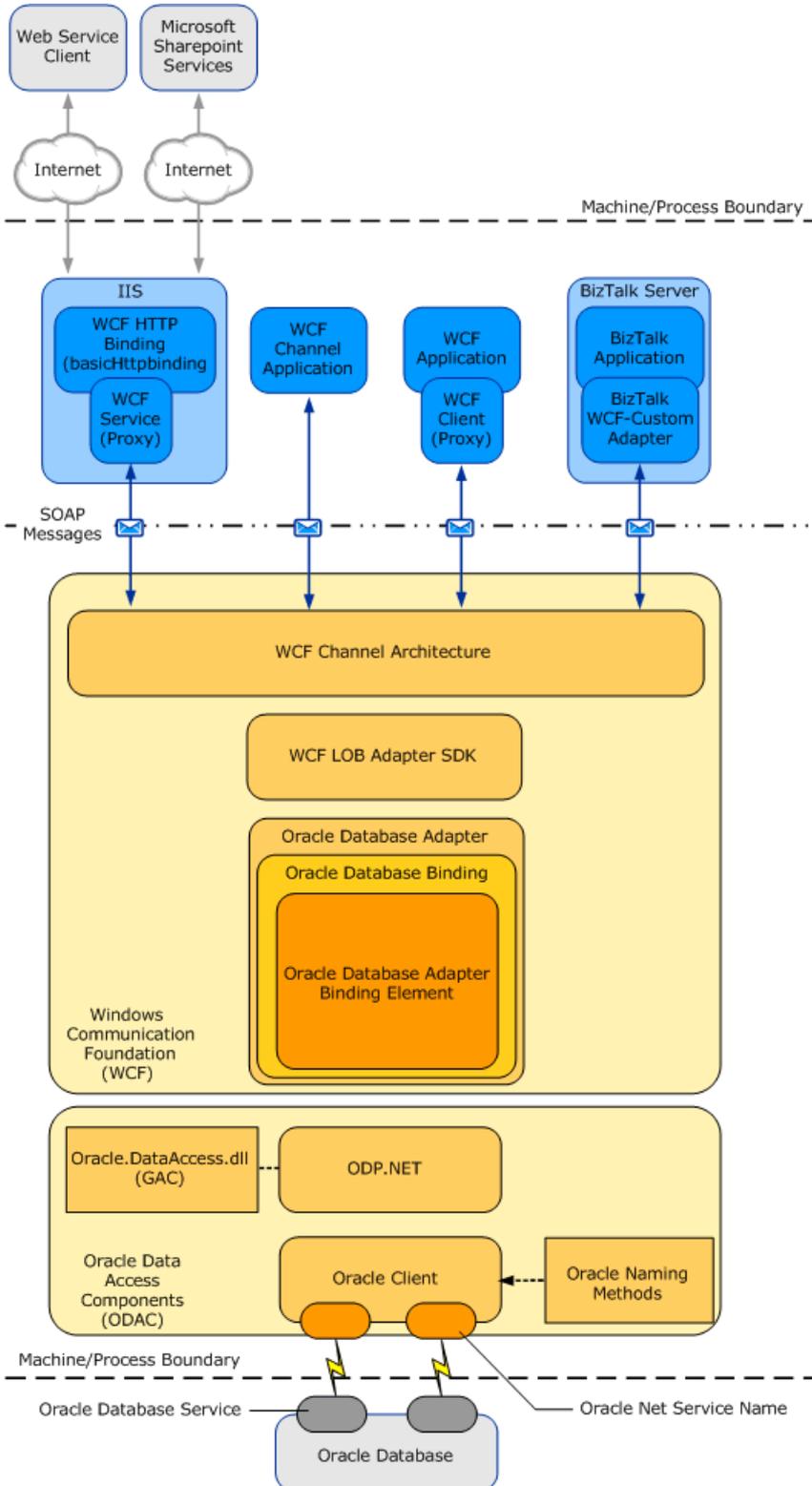
In This Section

- [Architecture for BizTalk Adapter for Oracle Database](#)
- [BizTalk Adapter for Oracle Database and the WCF LOB Adapter SDK](#)

Architecture for BizTalk Adapter for Oracle Database

The Microsoft BizTalk Adapter for Oracle Database is a Windows Communication Foundation (WCF) custom binding. This binding contains a single custom transport binding element that enables communication with an Oracle database. The Oracle Database adapter is wrapped by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK runtime and is exposed to applications through the WCF channel architecture. The Oracle Database adapter communicates with the Oracle database through the Oracle Data Provider for .NET (ODP.NET) and the Oracle client, which are part of the Oracle Data Access Components (ODAC) for Windows.

The following figure shows the end-to-end architecture for solutions that are developed by using the Oracle Database adapter.



Consuming the Adapter

The Oracle Database adapter exposes the Oracle database as a WCF service to client applications. To perform operations and access data on the Oracle database, client applications exchange SOAP messages with the Oracle Database adapter through

WCF channels. The preceding figure shows four ways in which the Oracle Database adapter can be consumed.

- Through a WCF channel model application. A WCF channel model application performs operations on the Oracle database by using the WCF channel model to exchange SOAP messages directly with the Oracle Database adapter. For more information about developing solutions for the Oracle Database adapter by using the WCF channel model, see [Developing Applications by Using the WCF Channel Model](#).
- Through a WCF service model application. A WCF service model application calls methods on a WCF client to perform operations on the Oracle database. A WCF client models the operations exposed by the Oracle Database adapter as .NET methods. You can use the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK or the WCF ServiceModel Metadata Utility Tool (svcutil.exe) to create a WCF client class from metadata exposed by the Oracle Database adapter. For more information about the WCF service model and the Oracle Database adapter, see [Developing Applications by Using the WCF Service Model](#).
- Through a BizTalk receive location or send port that is configured to use the Microsoft BizTalk WCF-Custom adapter. The WCF-Custom adapter enables the use of WCF extensibility features. By using the WCF-Custom adapter you can select and configure the Oracle DB Binding and the behavior for the receive location or send port. For more information about how to use the Oracle Database adapter in BizTalk Server solutions, see [Developing BizTalk Applications](#).
- Through an IIS-hosted Web service. In this scenario, a WCF service proxy generated by using the adapter is hosted in IIS using the standard WCF Http Binding. This exposes the service contract as a Web service to external users. IIS automatically hosts the adapter at runtime, which, in turn, communicates with the Oracle database.

The Oracle Database adapter and ODAC are always hosted in-process with the application or service that consumes the adapter.

The Oracle Database Adapter and WCF

WCF presents a programming model based on the exchange of SOAP messages over channels between clients and services. These messages are sent between endpoints exposed by a communicating client and service. An endpoint consists of:

- An *endpoint address*, which specifies the location at which messages are received
- A *binding*, which specifies the communication protocols used to exchange messages
- A *contract*, which specifies the operations and data types exposed by the endpoint.

A binding consists of one or more binding elements that stack on top of each other to define how messages are exchanged with the endpoint. At a minimum, a binding must specify the transport and encoding that are used to exchange messages with the endpoint. Message exchange between endpoints occurs over a channel stack that is composed of one or more channels. Each channel is a concrete implementation of one of the binding elements in the binding configured for the endpoint. For more information about WCF and the WCF programming model, see the WCF Help documentation at <http://go.microsoft.com/fwlink/?LinkId=196850>.

The Microsoft BizTalk Adapter for Oracle Database exposes a WCF custom binding, the Oracle DB Binding (**Microsoft.Adapters.OracleDB.OracleDBBinding**). By default, this binding contains a single custom transport binding element, the Oracle DB Adapter Binding Element (**Microsoft.Adapters.OracleDB.OracleDBAdapter**), which enables operations on an Oracle database.

Microsoft.Adapters.OracleDB.OracleDBBinding (the Oracle DB Binding) and **Microsoft.Adapters.OracleDB.OracleDBAdapter** (the Oracle DB Adapter Binding Element) are public classes and are also exposed to the configuration system. Because the Oracle DB Adapter Binding Element is exposed publicly, you can build your own custom WCF bindings capable of extending the functionality of the Oracle Database adapter. For example, you could implement a custom binding to support Enterprise Single Sign-on (SSO) in a WCF channel or service model solution. The reasons for doing this would be to aggregate database operations into a single multifunction operation or to perform schema transformation between operations implemented by a custom application and operations on the Oracle database.

The Oracle Database adapter is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and runs on top of the WCF LOB Adapter SDK runtime. The WCF LOB Adapter SDK provides a software framework and tooling infrastructure that the Oracle Database adapter uses to provide a rich set of features to users and adapter clients.

ODP.NET

The Oracle Database adapter connects with the Oracle database through the ODP.NET and the Oracle client. Both of these components are part of Oracle Data Access Components (ODAC).

ODP.NET implements a data provider for the Oracle database that is consistent with the ADO.NET interface. The Oracle Database adapter uses the classes exposed by ODP.NET to operate on the Oracle database.

The Oracle client provides connectivity to the Oracle database. You establish a connection to an Oracle database by providing a connection URI to the Oracle Database adapter. You can specify the connection URI in two ways:

- **Using tnsnames.ora.** In this approach, the connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, port number, etc. from the net service name entry in the file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.
- **Without using tnsnames.ora.** In this approach, the adapter clients specify the connection parameters directly in the connection URI. This does not require the net service name to be present in the tnsnames.ora file on the client computer. This approach does not even require the tnsnames.ora file to be present on the client computer.

For more information about the connection URI, see [Establishing a Connection to the Oracle Database](#). For more information about installing ODAC, see [Installing the BizTalk Adapter Pack](#).

See Also

Other Resources

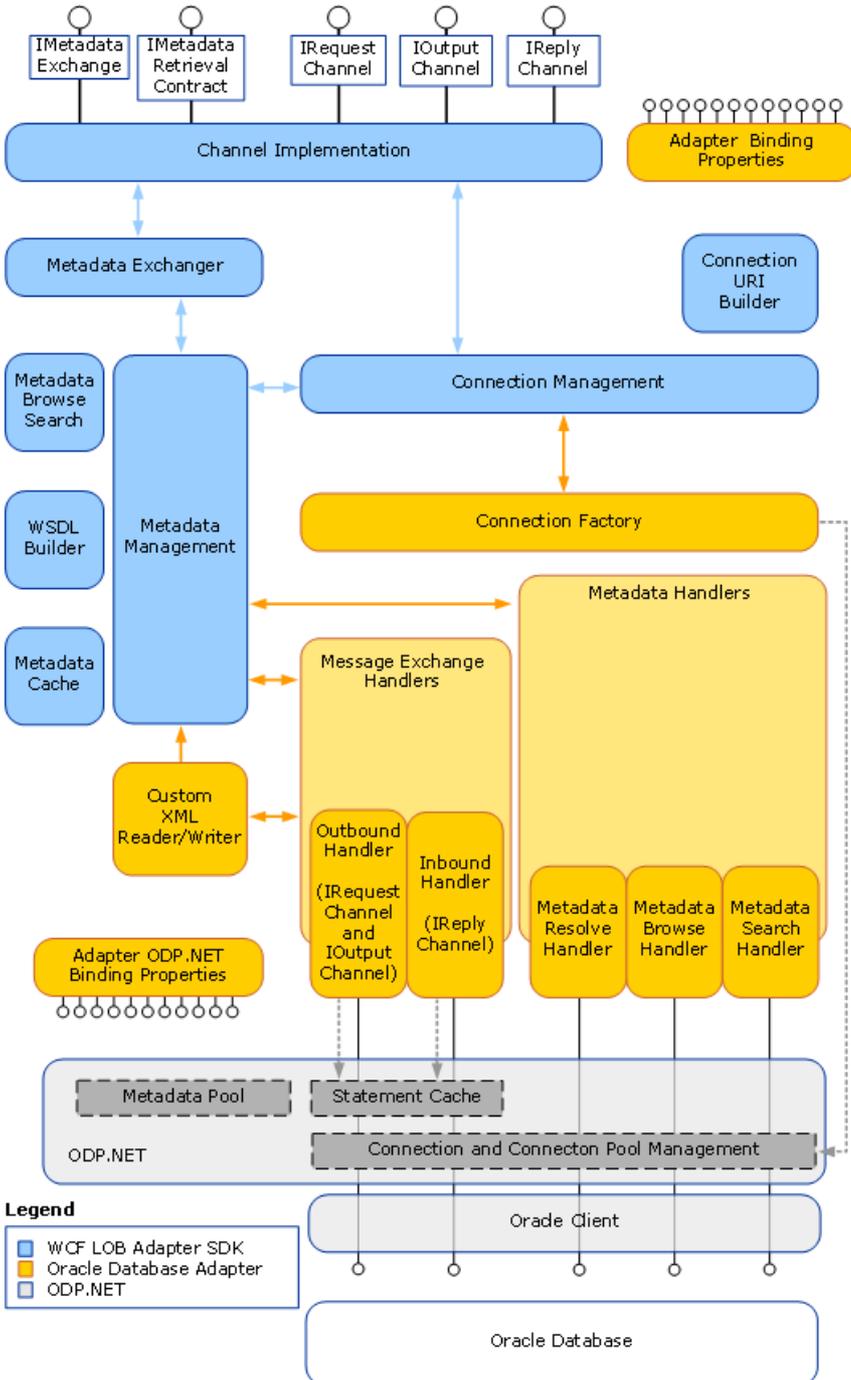
[Planning and Architecture](#)

[Understanding BizTalk Adapter for Oracle Database](#)

BizTalk Adapter for Oracle Database and the WCF LOB Adapter SDK

The Microsoft BizTalk Adapter for Oracle Database implements a set of core components that leverage functionality provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and provide connectivity to the Oracle database through Oracle Data Provider for .NET (ODP.NET).

The WCF LOB Adapter SDK serves as the software layer through which the Oracle Database adapter interfaces with the Windows Communication Foundation (WCF). ODP.NET serves as the layer through which the Oracle Database adapter interfaces with the Oracle database. The following figure shows the relationships between the internal components of the Oracle Database adapter, WCF LOB Adapter SDK, and ODP.NET.



See Also

Other Resources

[Planning and Architecture](#)

Security and Protection

A database often contains sensitive business information such as customer account details. Applications that use the Microsoft BizTalk Adapter for Oracle Database to access and modify this information either locally or across a distributed network might inadvertently expose it to access by unauthorized actors, unless efforts are made to protect and secure the data during transmission. Data protection and security are usually thought of in the following terms:

- *Authorization* controls access to a resource based on the identity of the requester.
- *Authentication* provides mechanisms for verifying the identity of a requester.
- *Data confidentiality* provides mechanisms for protecting the privacy of data through encryption.
- *Data integrity* provides mechanisms to digitally sign data, so that the receiver can ensure that the data has not been altered in-transit.

Another important area of concern is the user-name password credentials that you supply to the Oracle Database adapter. The adapter uses these credentials to open connections to the Oracle database. These credentials can be supplied in the connection URI; however, because the user name and password are clear text, the Oracle Database adapter provides alternative methods that you can use to supply these credentials in a more secure manner.

The topics in this section provide guidelines to help you better secure the solutions that you develop with the Oracle Database adapter.

In This Section

- [Security Considerations Between the Oracle Database and the Adapter](#)
- [Security Considerations When Using the Adapter with BizTalk Server](#)
- [Security Considerations When Programming on the Adapter](#)
- [Best Practices](#)

Security Considerations Between the Oracle Database and the Adapter

The Oracle Database adapter provides no support for helping to secure communication between it and the Oracle database. You must provide a security mechanism to help ensure appropriate levels of authorization, authentication, data privacy, and data integrity for data exchanges between the adapter and the Oracle database.

One possible mechanism for helping to provide more security across the network is Internet Protocol Security (IPsec). IPsec is a framework of open standards for protecting communications over Internet Protocol (IP) networks. For more information about IPsec and about using IPsec with Microsoft products, see the Microsoft TechNet article "IPsec" at <http://go.microsoft.com/fwlink/?LinkId=196851>.

However, in the absence of security mechanisms like IPsec, the administrator must configure native Oracle data encryption and integrity to ensure secure data exchanges between the adapter client and the Oracle database. For detailed information about configuring native Oracle data encryption and integrity, see <http://go.microsoft.com/fwlink/?LinkId=140032>.

You must supply user name password credentials to the Oracle Database adapter. The Oracle Database adapter uses these credentials to authenticate the user on the Oracle database when it opens a connection. These credentials provide a level of authorization on the Oracle database for the connection.

Note

The credentials used by the Oracle Database adapter to establish a connection on the Oracle database do not provide message-level or transport-level authentication or authorization for data traveling across the network. They are only used to open a connection and authenticate the user on the Oracle database.

The Oracle Database adapter provides a number of methods through which you can supply these credentials. For information about how to more securely provide Oracle credentials in BizTalk solutions, see [Security Considerations When Using the Adapter with BizTalk Server](#). For information about how to more securely provide Oracle database credentials in programming solutions, see [Security Considerations When Programming on the Adapter](#).

Managing Audit Logs

Audit logs enable you to store information about the actions performed by various clients on your enterprise software, and helps usage monitoring and problem tracking. However, the Oracle Database adapter does not provide any way to manage audit logs for the actions performed by the adapter clients on the Oracle database. This might pose a security threat as the adapter clients can repudiate the actions performed by them on the Oracle database. To mitigate this issue, you must enable audit trail in Oracle to log the actions performed by the adapter clients on the Oracle database.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Using the Adapter with BizTalk Server

When you configure a send port or a receive port (location) by using the BizTalk Server Administration console or use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas for a BizTalk solution, you must provide credentials for the Oracle database. It is important to supply these credentials in a secure way to help prevent them from being revealed to potentially malicious actors. This topic discusses how to most securely supply credentials for the Microsoft BizTalk Adapter for Oracle Database for BizTalk Server solutions.

A more general discussion of security in the context of BizTalk solutions is an expansive topic and is beyond the scope of this documentation. For information about how you can make your BizTalk solutions more secure, see the "Security and Protection" topic in the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101135>.

How Do I Protect Credentials When I Use the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard?

When you use the Consume Adapter Service Add-in to retrieve message schemas for a BizTalk solution, you must supply a user name and password for the Oracle database. You should only do this from the **Security** tab on the **Configure Adapter** dialog box. This ensures that your credentials will not be displayed in the **Configure a URI** field of the Consume Adapter Service Add-in dialog box, where anyone with access to your computer screen can read them. For more information about how to retrieve message schemas by using the Consume Adapter Service Add-in, including how to enter a user name and password for the Oracle database, see [Retrieving Metadata for Oracle Operations in Visual Studio](#).

How Do I Protect Credentials When I Configure a Send Port or a Receive Location?

BizTalk solutions use the Microsoft BizTalk WCF-Custom adapter to consume WCF services. The Oracle Database adapter is a WCF custom binding that enables clients to consume the Oracle database as if it were a WCF service. BizTalk solutions consume the Oracle Database adapter through send ports and receive locations that are configured to use the WCF-Custom adapter, which is, in turn, configured to use the Oracle Database adapter as its transport. For more information about how to configure send ports and receive ports (receive locations), including how to configure the WCF-Custom adapter, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

You configure the Oracle database credentials from the **Credentials** tab of the **WCF-Custom Transport Properties** dialog box for send ports or from the **Other** tab of the **WCF-Custom Transport Properties** dialog box for receive locations. Because the WCF-Custom adapter supports Enterprise Single Sign-On (SSO), you can choose to provide either a user name and password or an SSO affiliate application on either of these tabs. The following topics discuss both options.

User Name Password Credentials

You should only supply a user name and password from the **Credentials** tab (for send ports) or the **Other** tab (for receive locations) in the **WCF-Custom Transport Properties** dialog box. This ensures the following:

- Your credentials will not be displayed in the **Address (URI)** field of the dialog box. This prevents those who have access to your screen (or who have permissions that enable them to view the send port or receive location properties) from seeing your credentials.
- Your password will not be written to the binding file if you export the send port or receive port binding. This prevents anyone from with access to the file from viewing your password.

Enterprise Single Sign-On and SSO Affiliate Applications

You can configure the WCF-Custom adapter to use Enterprise Single Sign-on (SSO) to get the credentials for the Oracle database. SSO uses a database and a master secret to encrypt and store user credentials. It also provides services to map Microsoft Windows accounts to secondary credentials that are used to access a back-end system. By using SSO, you can map a Windows account to a user name and password on the Oracle database.

SSO uses *affiliate applications* and *SSO mappings* to map credentials to the back-end system. An affiliate application is a logical entity in SSO that refers to a system or an application that requires secondary credentials. An SSO mapping is associated with an affiliate application. It maps a Windows account to the secondary credentials used by that account to access the affiliate system or application. An SSO mapping can be associated with a Windows user account or with a group.

To use SSO with the Oracle Database adapter, you must do the following.

1. Create an affiliate application in SSO to hold the user name password credentials for the Oracle database. This step is often performed by someone with special types of SSO administrative privileges.
2. Create a user or group mapping for the affiliate application that maps your Windows account to the user name and password that are used to establish a connection with the Oracle database. Depending on your installation, a user might be able to perform this step or it might require someone with special types of SSO administrative privileges.

Note

When configured for SSO, the WCF-Custom adapter uses services provided by SSO to get the Oracle user name and password from the SSO database. It provides these (unencrypted) to the Oracle Database adapter, so that the adapter can open a connection to the Oracle database. SSO provides no encryption or protection across the connection between the Oracle Database adapter and the Oracle database.

For information about how to use SSO, including information about how to create affiliate applications and SSO mappings, see the "Using SSO" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103052>. For more general information about SSO, see the "Implementing Enterprise Single Sign-On" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103054>.

The AcceptCredentialsInUri Binding Property

The Oracle Database adapter surfaces the **AcceptCredentialsInUri** binding property. This property determines whether Oracle database credentials are permitted in the connection URI. By default, **AcceptCredentialsInUri** is **false** and the Oracle Database adapter throws an exception if credentials are included in the URI.

This property is surfaced because there are certain programming scenarios that require the credentials to be present in the connection URI. This should never be the case when you are configuring a send port or a receive location, or when you are using the Consume Adapter Service Add-in to retrieve message schemas from the Oracle Database adapter. It is recommended that you do not set **AcceptCredentialsInUri** to **true**. For more information about the Oracle Database adapter binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

The **AcceptCredentialsInUri** binding property is not available in BizTalk Server in the **Binding** tab while configuring a WCF-Custom or WCF-OracleDB receive or send port. To set the value of the **AcceptCredentialsInUri** binding property, you must open the adapter bindings file (XML file) that is created after you have generated metadata using the Consume Adapter Service Add-in, and then locate this binding property in the file. Specify an appropriate value for this binding property, save the binding file, and then import the binding file in BizTalk Server. See [Importing Bindings](#) for instructions.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Programming on the Adapter

How Do I Protect Credentials When I Use the Add Adapter Service Reference Visual Studio Plug-in?

When you use the Add Adapter Service Reference Plug-in to create a WCF client, you must supply a user name and password for the Oracle database. You should only do this from the **Security** tab on the **Configure Adapter** dialog box. By entering the Oracle credentials from the **Security** tab instead of directly into the **Configure a URI** field, you ensure the following:

- The credentials will not be displayed in the **Uri** field of the Add Adapter Service Reference Plug-in dialog box where anyone with access to your computer screen can read them.
- The credentials will not appear in the configuration file that the Add Adapter Service Reference Plug-in generates.

For more information about how to generate a WCF client by using the Add Adapter Service Reference Plug-in, including how to enter a user name and password for the Oracle database, see [Retrieving Metadata for Oracle Operations in Visual Studio](#).

What Are Best Practices for Setting Credentials in Code?

WCF provides the **ClientCredentials** class to help you configure the credentials that a client communication object, such as a **ChannelFactory**, uses to authenticate itself with a service. By using the **ClientCredentials** class, you ensure that WCF takes whatever authentication mechanisms are specified in that object's channel stack and applies them to the exchange between your client and the service.

Because the Oracle Database adapter is hosted in-process with its consuming application, it is not imperative to use the **ClientCredentials** class to set credentials on the client communication objects that the consuming application uses. It is, however, considered good practice to do so.

The Oracle Database adapter encourages the use of the **ClientCredentials** class. This property specifies whether the adapter will accept the user name and password for the Oracle database in the connection URI. **AcceptCredentialsInUri** defaults to **false**, which means that the adapter will throw an exception if the connection URI contains credentials. You can set **AcceptCredentialsInUri** to **true** to supply credentials in the connection URI.

The following example shows how to use the **Credentials** property to set credentials for the Oracle database on a **ChannelFactory**.

```
// Create binding and endpoint
OracleDBBinding binding = new OracleDBBinding();
EndpointAddress endpointAddress = new EndpointAddress("oracleDB://Adapter");

// Create the channel factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, endpointAddress);

// Set user name and password
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";

// Open the channel factory
factory.Open();
```

The following example shows how to use the **ClientCredentials** class to set credentials for the Oracle database on a WCF client.

```
// Initialize a new client for the SQLEXECUTE operation from configuration
SQLEXECUTEClient sqlExecuteClient = new SQLEXECUTEClient("OracleDBBinding_SQLEXECUTE");

// Set user name and password
sqlExecuteClient.ClientCredentials.UserName.UserName = "SCOTT";
sqlExecuteClient.ClientCredentials.UserName.Password = "TIGER";

// Open the client
sqlExecuteClient.Open();
```

How Can I Provide for More Secure Data Exchange Across Process Boundaries?

The Oracle Database adapter is hosted in-process with the application or service that consumes it. Because the adapter is hosted in-process with the consumer, there is no need to provide security on messages exchanged between the consumer and the Oracle Database adapter. However, if the consuming application or service sends messages that contain sensitive database information across a process boundary to another service or client, you should take measures to provide adequate protection for this data in your environment. Windows Communication Foundation (WCF) provides many options for helping to secure messages sent between clients and services. For more information about helping to secure messages sent between clients and services in WCF, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>. For more general information about security features that WCF provides, see "Windows Communication Foundation Security" at <http://go.microsoft.com/fwlink/?LinkId=89726>.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Best Practices

This section provides best practices that you should follow to more completely protect sensitive data when you use or develop applications that consume the Microsoft BizTalk Adapter for Oracle Database.

Security Best Practices for the Connection between the Oracle Database Adapter and the Oracle Database

- The Oracle Database adapter provides no support for helping to secure communication between it and the Oracle database. You must provide a mechanism to help ensure an adequate level of security for data exchanged between the adapter and the Oracle database.
- Do not provide user name password credentials for the Oracle database in the connection URI. See the following sections for alternative methods of providing credentials to the Oracle Database adapter.
- The Oracle Database adapter also enables you to use Windows Authentication while connecting to the Oracle database to generate metadata and perform operations, either through Visual Studio or BizTalk Server. Before using Windows Authentication, you must perform the steps listed in [Connecting to the Oracle Database Using Windows Authentication](#).

For more information, see [Security Considerations Between the Oracle Database and the Adapter](#).

Security Best Practices for Consuming the Oracle Database Adapter with BizTalk Server

- Do not provide user name password credentials for the Oracle database in the connection URI.
- When you use the Consume Adapter Service Add-in, enter the user name password credential for the Oracle database from the **Security** tab of the **Configure Adapter** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Oracle Database adapter on a send port, enter the user name password credential for the Oracle database from the **Credentials** tab of the **Configure WCF Custom Transport** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Oracle Database adapter on a receive location, enter the user name password credential for the Oracle database from the **Other** tab of the **Configure WCF Custom Transport** dialog box.
- The Oracle Database adapter also enables you to use Windows Authentication while connecting to the Oracle database to generate metadata and perform operations through BizTalk Server. Before using Windows Authentication, you must perform the steps listed in [Connecting to the Oracle Database Using Windows Authentication](#).

For more information, see [Security Considerations When Using the Adapter with BizTalk Server](#).

Security Best Practices for Consuming the Oracle Database Adapter with Programming Solutions

- It is sometimes necessary to provide the user name password credentials for the Oracle database in the connection URI; however, if possible, you should avoid doing this.
- When you use the Add Adapter Service Reference Visual Studio Plug-in, enter the user name password credential for the Oracle database from the **Security** tab of the **Configure Adapter** dialog box.
- In WCF Channel Model programming, use the **Credentials** property on the channel factory to set the user name password credential for the Oracle database.
- In WCF Service Model programming, use the **ClientCredentials** property on the WCF client to set the user name password credential for the Oracle database.
- If an application that consumes the Oracle Database adapter sends messages that contain sensitive database information across a process boundary to another service or client, ensure that these messages have sufficient security measures

applied to provide adequate data protection in your environment.

- The Oracle Database adapter also enables you to use Windows Authentication while connecting to the Oracle database to generate metadata and perform operations through Visual Studio. Before using Windows Authentication, you must perform the steps listed in [Connecting to the Oracle Database Using Windows Authentication](#).

For more information see, [Security Considerations When Programming on the Adapter](#).

Security Best Practices for Hosting the Oracle Database Adapter in IIS

Hosting the Oracle Database adapter in Microsoft Internet Information Services (IIS) as a Web service exposes operations surfaced by the Oracle Database adapter to Web clients. These operations might involve exchanging sensitive data over the Internet, so you should take measures to help ensure that this data is as secure as possible.

WCF provides two standard bindings for HTTP transport: the **BasicHttpBinding** provides basic HTTP transport with no security mechanisms; the **WSHttpBinding** supports both transport-level and message-level security mechanisms.

You can either use the **BasicHttpBinding** over an HTTPS connection, or use the **WSHttpBinding** to help protect your data. The WCF LOB Adapter SDK includes the WCF LOB Adapter Service Development Wizard to generate WCF service for LOB artifacts. This wizard only supports use of **BasicHttpBinding**.

You can also develop a custom HTTP binding to leverage additional security mechanisms that your environment provides. For more information about the security features that WCF provides, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>.

Security Best Practices for WCF Diagnostic Tracing and Message Logging

WCF supports diagnostic tracing and message logging. You configure diagnostic tracing and message logging either through configuration files or by using Windows Management Instrumentation (WMI). Depending on the configuration options you set, WCF diagnostic tracing or message logging can emit sensitive information to log files, where it could potentially be exposed to observation by unauthorized users.

Follow the recommendations provided in the WCF documentation to mitigate potential security threats exposed by enabling these features. At a minimum, you should observe the following best practices for diagnostic tracing and message logging:

- Do not enable "verbose" or "information" tracing in a production environment. This may lead to performance degradation. However, you must enable "warning" and "error" tracing in a production environment. If you enable tracing, you must take proper security measures to protect your data. See the WCF documentation for more information.
- Ensure that log files and configuration files are protected by access control lists (ACLs).

The following warnings apply specifically to the messages that are exchanged between a client application and the Oracle Database adapter:

- WCF diagnostic tracing can log the header (but not the body) of messages exchanged with the Oracle Database adapter. Because the message action is in the message header, this reveals the operations invoked on the Oracle Database adapter by the client.
- If WCF message logging is enabled and **logMessagesAtServiceLevel** is **true**, the message header (but not the message body) of messages exchanged between the adapter client and the Oracle Database adapter are logged. Because the message action is in the message header, this reveals the operations that the client invoked on the Oracle Database adapter. If **logEntireMessage** is also **true**, the message body will be logged. This can reveal sensitive database information.

For more information about improving security when you enable diagnostic tracing, see "Security Concerns and Useful Tips for Tracing" at <http://go.microsoft.com/fwlink/?LinkId=89796>. For more information about improving security when you enable message logging, see "Security Concerns for Message Logging" at <http://go.microsoft.com/fwlink/?LinkId=89797>.

See Also

Other Resources

[Security and Protection](#)

Development

The Microsoft BizTalk Adapter for Oracle Database is a Windows Communication Foundation (WCF) custom binding. Client applications can consume the Oracle Database adapter to invoke operations on Oracle database artifacts. The Oracle Database adapter can be consumed:

- Through a physical port binding in a BizTalk Server solution.
- By invoking methods on an instance of a .NET client proxy.
- As a hosted WCF service.
- By sending SOAP messages over a channel instance in code that uses the WCF channel model.

The following table:

- Lists the different operations that can be performed on an Oracle database using the Oracle Database adapter.
- Provides links to the topics containing information about performing the task using the chosen approach (BizTalk Server, WCF service model, or WCF channel model).

Task	BizTalk Server	WCF Service Model	WCF Channel Mod
Basic Insert, Update, Delete, and Select operations	Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server	Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model	Performing an Inser

<p>Operations on tables and views containing LOB data</p>	<p>Performing Operations on Tables with Large Object Types Data by Using BizTalk Server</p>	<p>Performing Operations on Tables with Large Object Types by Using the WCF Service Model</p>	
<p>Operations on functions and stored procedures</p>	<p>Invoking Functions and Procedures by Using BizTalk Server</p>	<p>Invoking Functions and Procedures by Using the WCF Service Model</p>	<p>Invoking a Function</p>
<p>Invoking overloaded functions</p>	<p>Invoking Overloaded Functions and Procedures by Using BizTalk Server</p>	<p>Invoking Functions and Procedures by Using the WCF Service Model</p>	

<p>O p e r a t i o n s o n f u n c t i o n s a n d p r o c e d u r e s W i t h R E F C U R S O R p a r a m e t e r s</p> <p>Invoking Functions and Procedures with REF CURSORS by Using BizTalk Server</p>	<p>Performing Operations Using REF CURSORS in the WCF Service Model</p>	
<p>O p e r a t i o n s o n f u n c t i o n s a n d p r o c e d u r e s w i t h R E C O R D t y p e s</p> <p>Invoking Functions and Procedures with RECORD Types by Using BizTalk Server</p>	<p>Performing Operations Using RECORD Types in the WCF Service Model</p>	

<p>O p e r a t i o n s o n t a b l e s a n d d i v i d e n t i f y i n g t h e m w i t h B I F L E d a t a t y p e s</p>	<p>Performing Operations on Tables with BFILE Data Types by Using BizTalk Server</p>		
<p>S Q L E X E C U T E o p e r a t i o n</p>	<p>Performing a SQLEXECUTE Operation by Using BizTalk Server</p>	<p>Performing a SQLEXECUTE Operation by Using the WCF Service Model</p>	<p>Performing a SQLEX</p>
<p>R e c e i v i n g p o l l i n g - b a s e d d a t a - c h a n g e m e s s a g e s</p>	<p>Polling Oracle Database by Using BizTalk Server</p>	<p>Receiving Polling-based Data-changed Messages by Using the WCF Service Model</p>	<p>Receiving Polling-ba</p>

P e r f o r m i n g C o m p o s i t e O p e r a t i o n s o n t h e O r a c l e D a t a b a s e	Performing Composite Operations on Oracle Database by Using BizTalk Server		
R e c e i v i n g D a t a b a s e C h a n g e N o t i f i c a t i o n s	Receiving Database Change Notifications Using BizTalk Server	Receiving Database Change Notifications Using the WCF Service Model	

The topics in this section provide information, procedures, and examples to help you develop applications that consume the Oracle Database adapter in both BizTalk Server and .NET programming solutions. The topics also provide information on other key aspects of using the adapters, such as:

- [Connecting to the Oracle database.](#)
- [Retrieving metadata from the Oracle database.](#)
- [Using binding properties to configure the adapter.](#)

In This Section

- [Establishing a Connection to the Oracle Database](#)
- [Retrieving Metadata for Oracle Operations in Visual Studio](#)
- [Working with BizTalk Adapter for Oracle Database Binding Properties](#)
- [Streaming and the Oracle Database Adapter](#)
- [Receiving Polling-based Data-changed Messages](#)
- [Developing BizTalk Applications](#)
- [Developing Applications by Using the WCF Service Model](#)
- [Developing Applications by Using the WCF Channel Model](#)
- [Retrieving Metadata Programmatically from the Oracle Database](#)
- [Using the Oracle Database Adapter with Microsoft Office SharePoint Server](#)

- [Samples](#)
- [How Do I?](#)

Establishing a Connection to the Oracle Database

The Microsoft BizTalk Adapter for Oracle Database is a Windows Communication Foundation (WCF) custom binding. As such, it enables communication to an Oracle database through a WCF endpoint address. In WCF, the endpoint address is typically expressed as a Uniform Resource Identifier (URI), which identifies the network location of the service. The Oracle Database adapter expresses this location as a connection URI, which contains properties that the Oracle Database adapter uses to establish a connection to the Oracle database.

You must specify a connection URI when you:

- Create a channel factory or a channel listener using the WCF channel model or when you create a WCF client or service host using the WCF service model.
- Create a physical port binding in a BizTalk Server solution.
- Use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class or WCF service interface for a WCF service model solution.
- Use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas from the Oracle Database adapter for a BizTalk Server solution.
- Use the ServiceModel Metadata Utility tool (svcutil.exe) to generate a WCF client class or WCF service interface for a WCF service model solution.

The Oracle Database adapter supports two ways of establishing a connection to the Oracle database:

- **Using tnsnames.ora.** In this approach, the connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, port no, etc. from the net service name entry in the file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.

◆ Important

Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the [connection URI](#) cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

- **Without using tnsnames.ora.** In this approach, the adapter clients specify the connection parameters directly in the connection URI. This does not require the net service name to be present in the tnsnames.ora file on the client computer. This approach does not even require the tnsname.ora file to be present on the client computer.

◆ Important

This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

The topics in this section describe how to establish a connection between the Microsoft BizTalk Adapter for Oracle Database and the Oracle database by providing you with:

- Information about configuring the Oracle client.
- Information about the connection properties and the structure of the Oracle connection URI.
- Links to topics that show how to establish a connection by using the Oracle Database adapter.
- Information about connecting to the Oracle database using Windows Authentication.

In This Section

- [Configuring the Oracle Client](#)
- [The Oracle Database Connection URI](#)
- [Connecting to the Oracle Database Using Windows Authentication](#)

See Also

Concepts

[Development](#)

Configuring the Oracle Client

◆ Important

This topic is relevant only if you are using tnsnames.ora to connect to the Oracle database.

The Microsoft BizTalk Adapter for Oracle Database connects to the Oracle database through the Oracle client installed on your computer. The Oracle Database adapter passes the net service name that you specify in the connection URI to the Oracle client to establish a connection to the Oracle database. The net service name is an alias that the Oracle client uses to acquire connection information for the target Oracle database service.

The Oracle client resolves the net service name according to the naming method that it is configured to use. You use the Oracle Net Configuration Assistant to configure the naming methods to be used by the Oracle client. The Oracle Database adapter supports the Local Naming method for connecting to the Oracle database. This method uses a local file, tnsnames.ora, to resolve the net service name.

The tnsnames.ora file associates net service names with connect descriptors that contain the information the Oracle client needs to establish a connection to a specific Oracle database service (instance). The following is a sample entry from tnsnames.ora.

```
ADAPTER =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = yourOracleServer)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = yourOracleDatabaseServiceName)
    )
  )
```

In this sample entry, ADAPTER is the net service name. The connect descriptor specifies address information and the service name of the Oracle database service associated with ADAPTER. You can use the Oracle Net Configuration Assistant to create and configure net service names in tnsnames.ora. After you have configured the net service name, you can specify it in a connection URI as in the following example.

```
oracledb://ADAPTER
```

For more information about using the Oracle Net Configuration Assistant and about tnsnames.ora, see the Oracle Database Net Services Administrator's Guide. Consult your database administrator about configuration details for your specific installation.

See Also

Other Resources

[Establishing a Connection to the Oracle Database](#)

The Oracle Database Connection URI

The Microsoft BizTalk Adapter for Oracle Database connection URI contains properties that the adapter uses to establish a connection to the Oracle database. This topic provides information about how to specify the connection URI to connect to the Oracle database using tnsnames.ora and without using tnsnames.ora. It also provides information about using the connection URI to connect to the Oracle database.

Connection URI to Connect to the Oracle Database Using tnsnames.ora

Important

- For this approach, you must add the net service name entry in the tnsnames.ora file on the computer with the adapter client installed. For more information about the net service name entry, see [Configuring the Oracle Client](#).
- Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the connection URI cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

A typical endpoint address URI in WCF is represented as: `scheme://userauthparams@hostinfoparams?query_string`, where:

- `scheme` is the scheme name.
- `userauthparams` is a name-value collection of parameters required for user authentication by the endpoint.
- `hostinfoparams` is information required to establish the connection to the host; for example, a path.
- `query_string` is an optional name-value collection of parameters delimited by a question mark (?).

The Oracle Database adapter connection URI adheres to this basic format and is implemented as follows:

```
oracledb://User=[USER_NAME];Password=[PASSWORD]@[NET_SERVICE_NAME]?PollingId=[POLLING_ID]
```

The following table explains the properties contained in the connection URI.

Connection URI Property	Category	Description
[USER_NAME]	userauthparams	<p>The user name to use for authentication on the Oracle database; for example, SCOTT. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI.</p> <p>Note The Oracle Database adapter preserves the case of the value that you enter for the user name when it opens a connection on the Oracle database. User names on the Oracle database are case-sensitive. You should ensure that you provide Oracle user names to the Oracle Database adapter in the case expected by your Oracle database. Typically, this means that the user name in the SCOTT/TIGER credential should be upper case: "SCOTT".</p>

[PASSWORD]	userauthparams	The password to use for authentication on the Oracle database; for example, TIGER. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI. 📌Note The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. For release 10g and earlier, passwords on the Oracle system are not case-sensitive.
[NET_SERVICE_NAME]	hostinfoparams	A net service name that is specified in the tnsnames.ora file on the computer where the Oracle Database adapter is installed. For more information about net service names and tnsnames.ora, see Configuring the Oracle Client .
[POLLING_ID]	query_string	An optional string that should be appended by the adapter to the standard namespace of the POLLINGSTMT operation. This enables you to specify a unique namespace for each polling operation when a project contains multiple polling operations. You do not have to specify a PollingId string if your project contains only one POLLINGSTMT operation.

📌Note
Query parameters are also used in the connection URI when an endpoint address is specified for a WCF Metadata Exchange client.

Connection URI to Connect to the Oracle Database Without Using tnsnames.ora

📌Important
<ul style="list-style-type: none"> For this approach, the net service name in the tnsnames.ora file, or the actual tnsnames.ora file itself does not need to be present on the client computer. This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

A typical endpoint address URI in WCF is represented as: `scheme://userauthparams@hostinfoparams?query_string`, where:

- `scheme` is the scheme name.
- `userauthparams` is a name-value collection of parameters required for user authentication by the endpoint.
- `hostinfoparams` is information required to establish the connection to the host; for example, server name, port number, etc.
- `query_string` is an optional name-value collection of parameters delimited by a question mark (?).

The Oracle Database adapter connection URI adheres to this basic format and is implemented as follows:

```
oracledb://User=[USER_NAME];Password=[PASSWORD]@[SERVER_NAME]:[PORT_NUMBER]/[SERVICE_NAME]/[SERVICE_TYPE]?PollingId=[POLLING_ID]
```

The following table explains the properties contained in the connection URI.

Connection URI Property	Category	Description
-------------------------	----------	-------------

[USER_NAME]	user authentication parameters	The user name to use for authentication on the Oracle database; for example, SCOTT. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI. 📌Note The Oracle Database adapter preserves the case of the value that you enter for the user name when it opens a connection on the Oracle database. User names on the Oracle database are case-sensitive. You should ensure that you provide Oracle user names to the Oracle Database adapter in the case expected by your Oracle database. Typically, this means that the user name in the SCOTT/TIGER credential should be upper case: "SCOTT".
[PASSWORD]	user authentication parameters	The password to use for authentication on the Oracle database; for example, TIGER. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI. 📌Note The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. For release 10g and earlier, passwords on the Oracle system are not case-sensitive.
[SERVER_NAME]	host information parameters	Name of the server on which the Oracle database is running. This is mandatory.
[PORT_NUMBER]	host information parameters	The Oracle Net Listener port. If no value is specified, the adapter takes the default value 1521.
[SERVICE_NAME]	host information parameters	The Oracle database service name. This is mandatory.
[SERVICE_TYPE]	host information parameters	The type of Oracle service. The possible values are Dedicated or Shared . A dedicated service uses a dedicated server process to serve only one user process. A shared service uses a shared server process that can serve multiple user processes. Default is Dedicated .
[POLLING_ID]	query string	An optional string that should be appended by the adapter to the standard namespace of the POLLINGSTMT operation. This enables you to specify a unique namespace for each polling operation when a project contains multiple polling operations. You do not have to specify a PollingId string if your project contains only one POLLINGSTMT operation.
📌Note		
Query parameters are also used in the connection URI when an endpoint address is specified for a WCF Metadata Exchange client.		

Oracle Database Credentials and the Connection URI

By default, the Oracle Database adapter throws an exception when the Oracle database credentials are specified in the connection URI. This is because these credentials are represented as plain text in the connection URI, and this poses a security risk. You can set the **AcceptCredentialsInUri** binding property to control whether the connection URI can contain credentials for the Oracle database. If the **AcceptCredentialsInUri** property is **false**, the Oracle Database adapter throws an exception if the connection URI contains Oracle database credentials; if the property is **true**, no exception is thrown. There are a few limited scenarios in which it is necessary to specify credentials in the connection URI; for example, to receive the inbound POLLINGSTMT operation when you use the WCF service model or the WCF channel model. For most situations, however, you

should avoid providing credentials in the connection URI. For more information about how to more securely provide credentials for the Oracle database, see [Security and Protection](#).

◆ Important

Due to the security risks posed by passing credentials in strings as plain text, you should avoid specifying Oracle database connection credentials in the connection URI.

Using Reserved Characters in the Connection URI

The Oracle Database adapter does not support specifying a connection URI that has special characters for any of the parameter values. If the connection parameter values contain special characters, make sure you do one of the following:

- If you are specifying the URI in Visual Studio using Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. If you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.
- If you are specifying the URI while creating a send or receive port in BizTalk Server Administration console, and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

Using the Connection URI to Connect to the Oracle Database

The following is an example of a connection URI for Oracle Database adapter.

Using tnsnames.ora	Without using tnsnames.ora
<pre>oracledb://ADAPTER</pre> <p>In this example, ADAPTER is a net service name that is associated with the SERVICE NAME and connection information for the target Oracle database in tnsnames.ora.</p>	<pre>oracledb://yourOracleServer:1521/yourOracleDatabaseServiceName/Dedicated</pre> <p>In this example, the server name is "yourOracleServer" and the service name is "yourOracleDatabaseServiceName".</p>

The following is an example of a connection URI for a POLLINGSTMT operation. This URI includes a PollingId parameter to modify the namespace of the POLLINGSTMT operation.

Using tnsnames.ora	Without using tnsnames.ora
<pre>oracledb://ADAPTER?PollingId=MyPollingNotification1</pre>	<pre>oracledb://yourOracleServer:1521/yourOracleDatabaseServiceName/Dedicated?PollingId=MyPollingNotification1</pre>

For the above connection URIs, the Oracle Database adapter creates the following namespace for the POLLINGSTMT operation.

```
http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTMyPollingNotification1
```

For information about how to establish a connection to the Oracle database when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in](#).
- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Specifying a Client Binding for the Oracle Database](#).

- Use the WCF ServiceModel Metadata Utility Tool (svcutil.exe), see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Oracle Database.](#)

See Also

Concepts

[Configuring the Oracle Client](#)

Other Resources

[Establishing a Connection to the Oracle Database](#)

Connecting to the Oracle Database Using Windows Authentication

The Microsoft BizTalk Adapter for Oracle Database enables adapter clients to use Windows Authentication to establish a connection with the Oracle database. To use Windows Authentication, the adapter clients must specify "/" for user name and leave the password blank. For more information about connecting to the Oracle database using Windows Authentication, see [Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in](#).

To enable adapter clients to use Windows Authentication to connect to an Oracle database, you must perform the following tasks on the computer running the Oracle database.

1. Make sure that the `sqlnet.ora` file on both the client and the server, available under `ORACLE_BASE\ORACLE_HOME\network\admin\sqlnet.ora`, has the following entry:

```
SQLNET.AUTHENTICATION_SERVICES= (NTS)
```

2. Connect to the Oracle database as SYSDBA.
3. Create the Windows user as an external user in the Oracle database. Note that the user name must be in upper case.

```
CREATE USER "OPS$<DOMAIN_NAME>\<USER_NAME>" IDENTIFIED EXTERNALLY;
```

4. Grant privileges to the user.

```
GRANT CONNECT,RESOURCE TO "OPS$<DOMAIN_NAME>\<USER_NAME>";
```

5. To enable the newly created user, logging in using Windows Authentication, to access the Oracle database artifacts, you can change the user's schema to the SCOTT schema. You can add the following SQL command to the logon script that changes the user's default schema to SCOTT when the user logs on.

```
alter session set current_schema=SCOTT;
```

6. Even though you changed the schema of the user to the SCOTT schema, you will still not be able to see the Oracle database artifacts while browsing and generating metadata using the Oracle Database adapter. This is because the newly created user does not have permissions for the SCOTT schema. Make sure you provided permission for the SCOTT schema to the newly created user.

See Also

Concepts

[Configuring the Oracle Client](#)

Other Resources

[Establishing a Connection to the Oracle Database](#)

Retrieving Metadata for Oracle Operations in Visual Studio

The Microsoft BizTalk Adapter for Oracle Database provides three Visual Studio components that you can use to help you develop solutions using the adapter.

- The Consume Adapter Service BizTalk Project Add-in and the Add Adapter Metadata Wizard are available in BizTalk Server projects. You use the Consume Adapter Service Add-in and the Add Adapter Metadata Wizard to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The Add Adapter Service Reference Visual Studio Plug-in is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

All these three Visual Studio components simplify development by:

- Providing a Microsoft Windows interface through which you can browse and search for operations that you want to use in your solution.
- Retrieving metadata exposed by the adapter for these target operations.
- Converting that metadata, which is expressed as a Web Services Description Language (WSDL) document by the adapter, into a form that you can use in your solution (XSD message schemas for BizTalk projects or a .NET object representation of a service contract for the WCF service model) and adding it to your project.

This section provides instructions about how to use Consume Adapter Service Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Plug-in.

In This Section

- [Connecting to the Oracle Database in Visual Studio](#)
- [Browsing, Searching, and Retrieving Metadata for Oracle Operations](#)

See Also

Concepts

[Development](#)

Connecting to the Oracle Database in Visual Studio

This section provides information about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

- The **Consume Adapter Service BizTalk Project Add-in** is available in BizTalk Server projects. You use the Consume Adapter Service Add-in to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The **Add Adapter Metadata Wizard** is available in BizTalk Server projects. You use the Add Adapter Metadata Wizard to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).

Note

Because the Oracle Database adapter is exposed both as a WCF-Custom binding and as BizTalk adapter, you can use either the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard from a BizTalk project to connect to SQL Server.

- The **Add Adapter Service Reference Visual Studio Plug-in** is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

To use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in, you must first connect to the Oracle database. All three methods present a dialog box through which you configure a connection by setting the following:

- **Connection parameters.** These are the parameters that are used to build the connection URI. You must specify a data source (Oracle net service name).
- **User name password credentials for the Oracle database.** These are used to authenticate you on the Oracle database when the connection is established. You must specify a user name and password.
- **Binding properties.** Binding properties are optional at design-time, that is, while generating metadata for operations. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

At a minimum, when you configure the connection to the Oracle database, you only have to specify binding properties and connection parameters that are needed to establish the connection and that affect the metadata returned by the Oracle Database adapter for the operations you want to target. However, you might also want to specify values for any additional binding properties and connection parameters that will be used at run time. This is because:

- The Consume Adapter Service Add-in creates a BizTalk port binding file from the binding properties and connection parameters that you specify when you configure the connection, and adds this file to your project. Later, you can use this binding file to create a port in the BizTalk Server Administration console. For more information about the binding file, see [Configuring a Physical Port Binding by Using a Port Binding File](#).
- The Add Adapter Service Reference Plug-in creates an app.config file from the binding properties and connection properties that you specify when you configure the connection, and adds this file in your project directory.

In This Section

- [Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in](#)
- [Connecting to the Oracle Database in Visual Studio Using Add Adapter Metadata Wizard](#)

- [Connecting to the Oracle Database in Visual Studio Using Add Adapter Service Reference Plug-in](#)

See Also

Other Resources

[Retrieving Metadata for Oracle Operations in Visual Studio](#)

Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in is installed when you install WCF LOB Adapter SDK. The Consume Adapter Service Add-in loads all the WCF-Custom bindings installed on the computer. To connect to the Oracle database using the WCF-based Oracle Database adapter in a BizTalk project, you must use the **oracleDBBinding**.

This topic provides instructions on how to use the Consume Adapter Service Add-in.

Connecting to an Oracle Database Using the Consume Adapter Service Add-in

Perform the following steps to connect to an Oracle database using the Consume Adapter Service Add-in.

To connect to an Oracle database

1. To connect using the Consume Adapter Service Add-in in a BizTalk solution:
 - a. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - b. In the **Add Generated Items** dialog box, do the following:

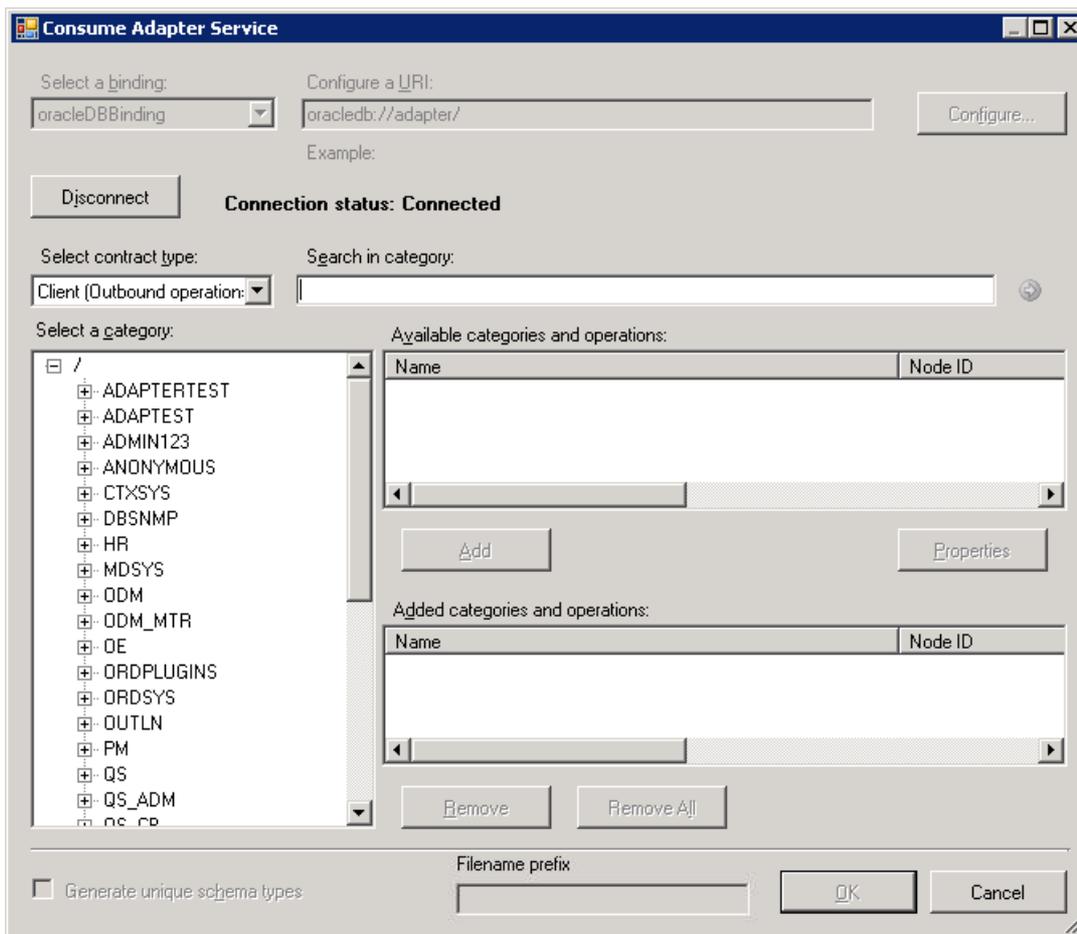
Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

- c. Click **Add**. The Consume Adapter Service Add-in opens.
2. From the **Select a binding** drop-down list, select **oracleDBBinding** and click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle database.
 - a. To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes. Make sure you adhere to the following considerations when specifying the user name and password to connect to an Oracle database:
 - **User name.** The Oracle Database adapter preserves the case of the value that you enter for the user name when it opens a connection on the Oracle database. User names on the Oracle database are case-sensitive. You should ensure that you provide Oracle user names to the Oracle Database adapter in the case expected by your Oracle database. Typically, this means that the user name in the SCOTT/TIGER credential should be upper case: "SCOTT".
 - **Password.** The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. For release 10g and earlier, passwords on the Oracle system are not case-sensitive.
 - b. To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
4. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle Database adapter, see [The Oracle Database Connection URI](#).
5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For example, if you want to target the POLLINGSTMT operation, you must set the **PollingStatement** binding property. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

6. Click **OK**.

7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



See Also

Concepts

[Connecting to the Oracle Database in Visual Studio](#)

[Connecting to the Oracle Database Using Windows Authentication](#)

Connecting to the Oracle Database in Visual Studio Using Add Adapter Metadata Wizard

The Oracle Database adapter is also exposed as a BizTalk adapter and, therefore, you can use the Add Adapter Metadata Wizard to generate schema for the operations you want to perform on the Oracle database using the adapter.

This topic provides instructions on how to use the Add Adapter Metadata Wizard.

Connecting to an Oracle Database Using the Add Adapter Metadata Wizard

Perform the following steps to connect to an Oracle database using the Add Adapter Metadata Wizard.

To connect to an Oracle database

1. To connect using the Add Adapter Metadata Wizard in a BizTalk solution:
 - a. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - b. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

- c. Click **Add**. The Add Adapter Metadata Wizard opens.
 - d. In the Add Adapter Metadata Wizard, select **WCF-OracleDB**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-OracleDB port configured in BizTalk, select the port from the Port list.

- e. Click **Next**.
2. From the **Select a binding** drop-down list, select **oracleDBBinding** and click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle database.
 - a. To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes. Make sure you adhere to the following considerations when specifying the user name and password to connect to an Oracle database:
 - **User name.** The Oracle Database adapter preserves the case of the value that you enter for the user name when it opens a connection on the Oracle database. User names on the Oracle database are case-sensitive. You should ensure that you provide Oracle user names to the Oracle Database adapter in the case expected by your Oracle database. Typically, this means that the user name in the SCOTT/TIGER credential should be upper case: "SCOTT".
 - **Password.** The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. For release 10g and earlier, passwords on the Oracle system are not case-sensitive.
 - b. To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.

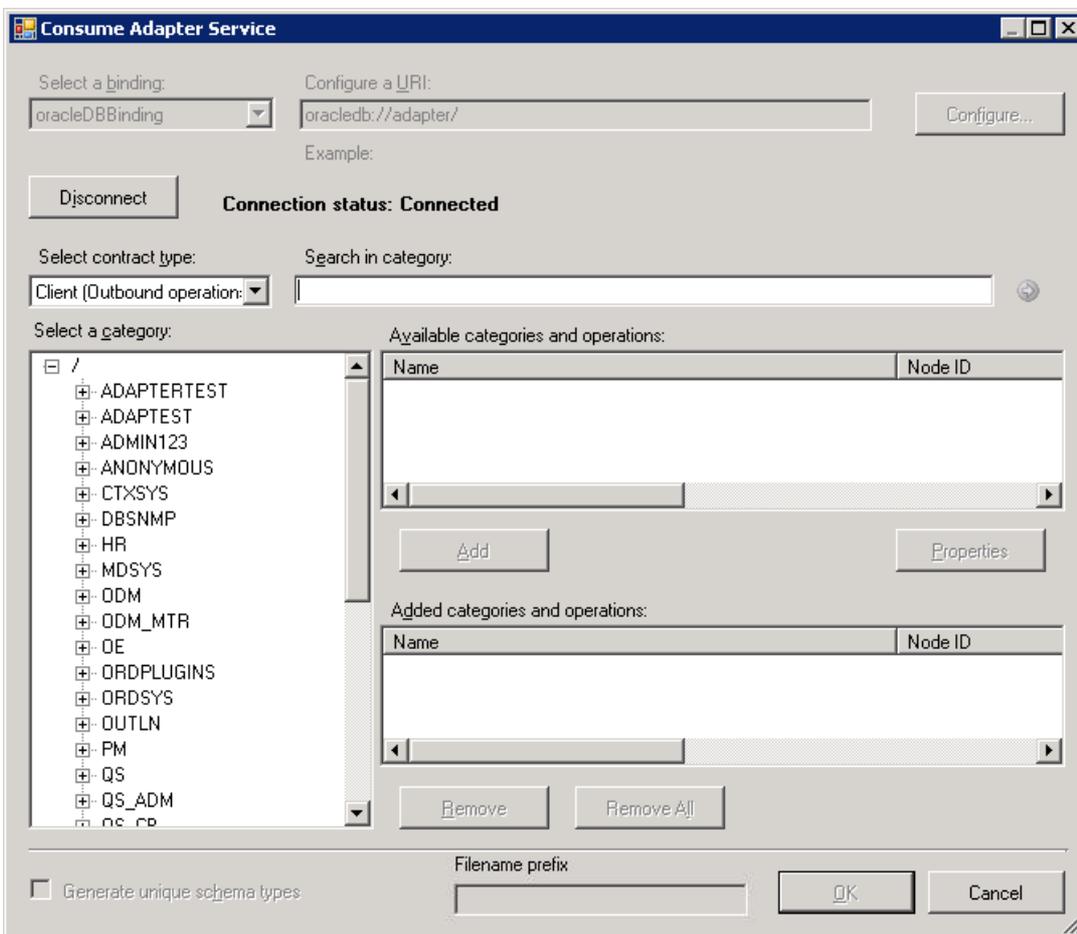
- Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle Database adapter, see [The Oracle Database Connection URI](#).
- Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For example, if you want to target the POLLINGSTMT operation, you must set the **PollingStatement** binding property. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

If you are generating metadata using Add Adapter Metadata Wizard and you selected an existing WCF-OracleDB send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

- Click **OK**.
- Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



See Also

Concepts

- [Connecting to the Oracle Database in Visual Studio](#)
- [Connecting to the Oracle Database Using Windows Authentication](#)

Connecting to the Oracle Database in Visual Studio Using Add Adapter Service Reference Plug-in

To connect to the Oracle database using the **adapteroracle_short** in a .NET programming solution, you must use the Add Adapter Service Reference Visual Studio Plug-in. This topic provides instructions on how to use the Add Adapter Service Reference Plug-in.

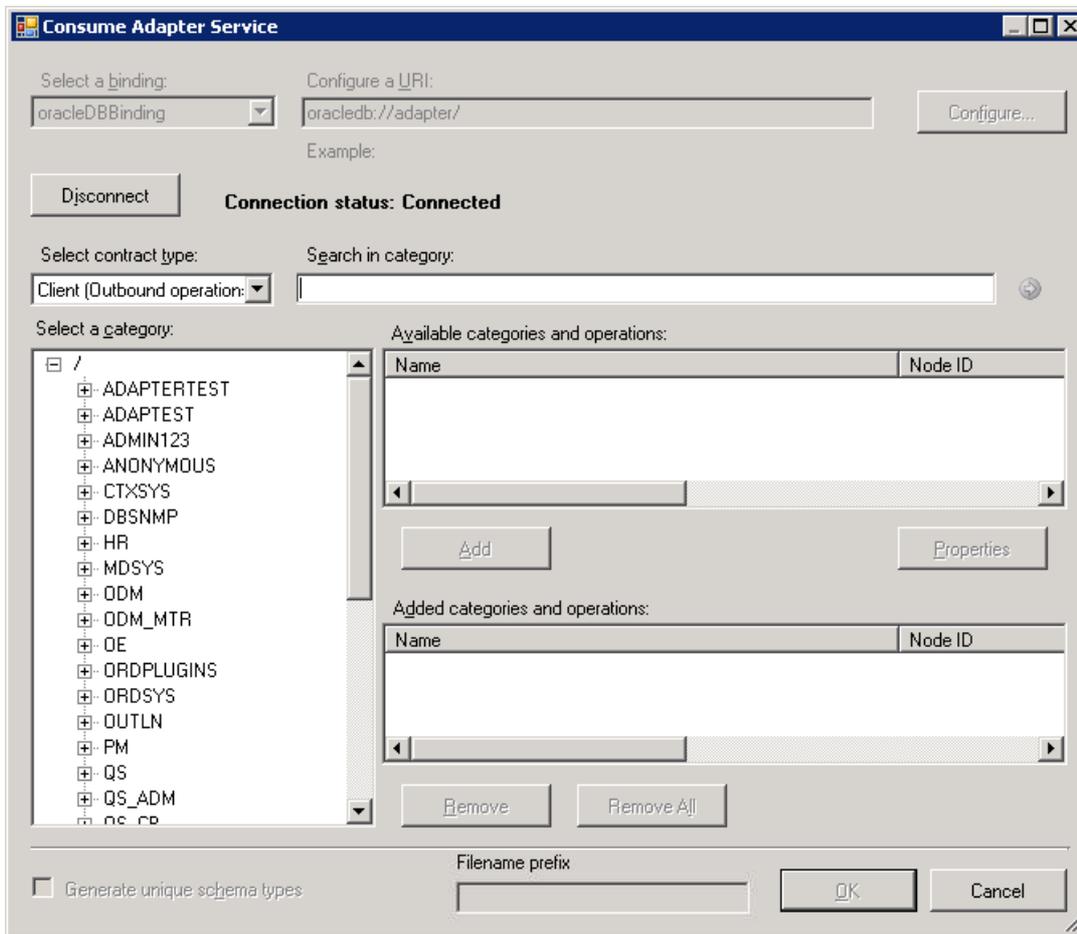
Connecting to an Oracle Database Using the Add Adapter Service Reference Plug-in

Perform the following steps to connect to an Oracle database using the Add Adapter Service Reference Plug-in.

To connect to an Oracle database

1. To connect using the Add Adapter Service Reference Plug-in in a programming solution:
 - a. Create a project using Visual Studio.
 - b. Right-click the project in Solution Explorer, and then click **Add Adapter Service Reference**. The Add Adapter Service Reference Plug-in opens.
2. From the **Select a binding** drop-down list, select **oracleDBBinding** and click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle database.
 - a. To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes. Make sure you adhere to the following considerations when specifying the user name and password to connect to an Oracle database:
 - **User name**. The Oracle Database adapter preserves the case of the value that you enter for the user name when it opens a connection on the Oracle database. User names on the Oracle database are case-sensitive. You should ensure that you provide Oracle user names to the Oracle Database adapter in the case expected by your Oracle database. Typically, this means that the user name in the SCOTT/TIGER credential should be upper case: "SCOTT".
 - **Password**. The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. For release 10g and earlier, passwords on the Oracle system are not case-sensitive.
 - b. To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
4. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle Database adapter, see [The Oracle Database Connection URI](#).
5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For example, if you want to target the POLLINGSTMT operation, you must set the **PollingStatement** binding property. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
6. Click **OK**.
7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



See Also

Concepts

[Connecting to the Oracle Database in Visual Studio](#)

[Connecting to the Oracle Database Using Windows Authentication](#)

Browsing, Searching, and Retrieving Metadata for Oracle Operations

This section provides information about how to use Consume Adapter Service BizTalk Project Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Visual Studio Plug-in. By using these Visual Studio components, you can:

- Browse for operations for which to retrieve metadata.
- Search for operations for which to retrieve metadata.
- Add message schemas for selected operations and port binding configuration files to a BizTalk server project when using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.
- Add a WCF client class or a WCF service contract (interface) for selected operations and a configuration file (app.config) to a non-BizTalk programming project when using the Add Adapter Service Reference Plug-in.

You must connect to the Oracle database before you can browse, search, or retrieve metadata for target operations. For information about how to connect to the Oracle database when you use Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in, see [Connecting to the Oracle Database in Visual Studio](#).

Note

- The Consume Adapter Service BizTalk Project Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Visual Studio Plug-in present essentially the same interface when you browse and search for operations, so all the three components are covered in the same topics.
- You can select category nodes to return all the operations in that category's subtree—for example, an entire table or schema (or even all the tables in a schema).

Browsing for Operations

While browsing metadata using Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in, the Oracle Database adapter surfaces:

- Operations that can be performed on tables, views, stored procedures, functions, and packages.
- The SQLEXECUTE operation, which enables adapter clients to execute any generic data manipulation language (DML) or stored procedure in an Oracle database.
- The POLLINGSTMT and Notification operations, which enable adapter clients to obtain inbound data from the Oracle database. It also exposes a list of stored procedures, functions and packages under the respective schemas that are exposed as operations for polling.

Note

- By using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in, you can browse category and operation nodes using a Windows interface.
- The SQLEXECUTE, POLLINGSTMT, and Notification operations are presented directly under the root node in the category tree. You must select the root node to view these outbound and inbound operations.

For more information about browsing metadata, see [How Does the Adapter Surface Oracle Metadata?](#)

Perform the following steps to browse for operations exposed for different artifacts in an Oracle database using the Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in.

To browse metadata in an Oracle database

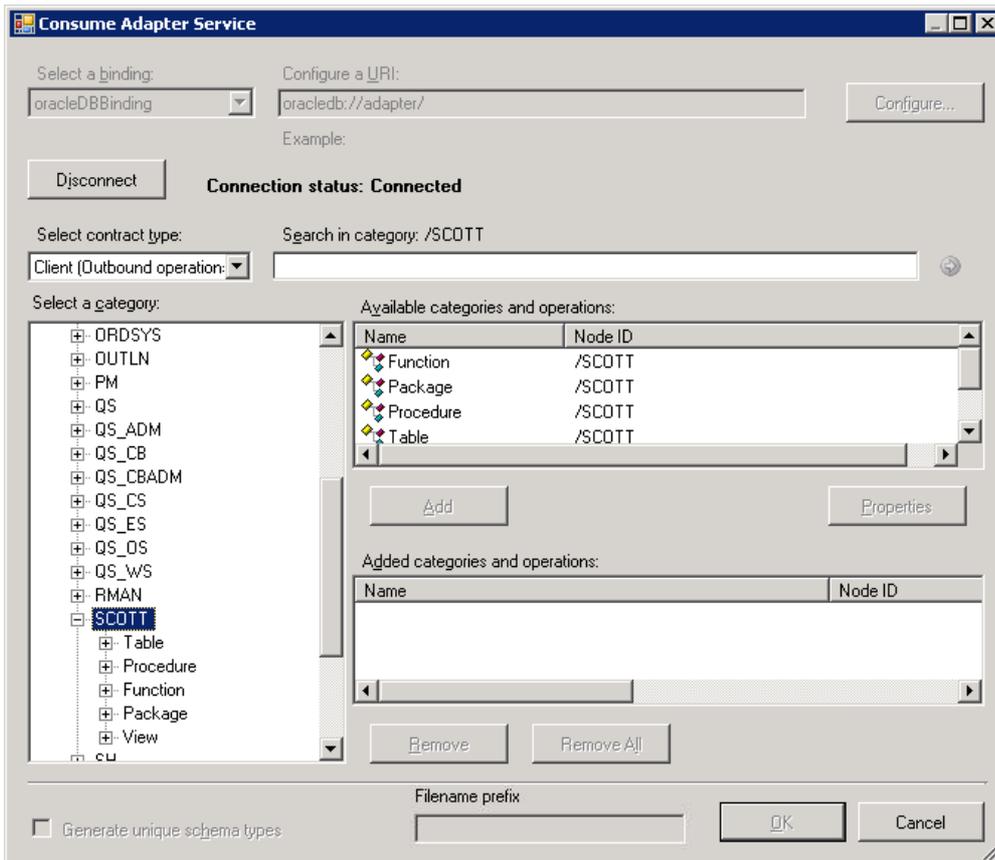
1. Connect to an Oracle database using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in. See [Connecting to the Oracle Database in Visual Studio](#) for instructions.

- From the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound or outbound operations using the adapter.
- The **Select a category** box lists the schemas in the Oracle database. Click a schema to see the tables, procedures, functions, packages, and views accessible to the schema in the **Available categories and operations** box. Alternatively, you can see the categorization by expanding the schema node.

Tip

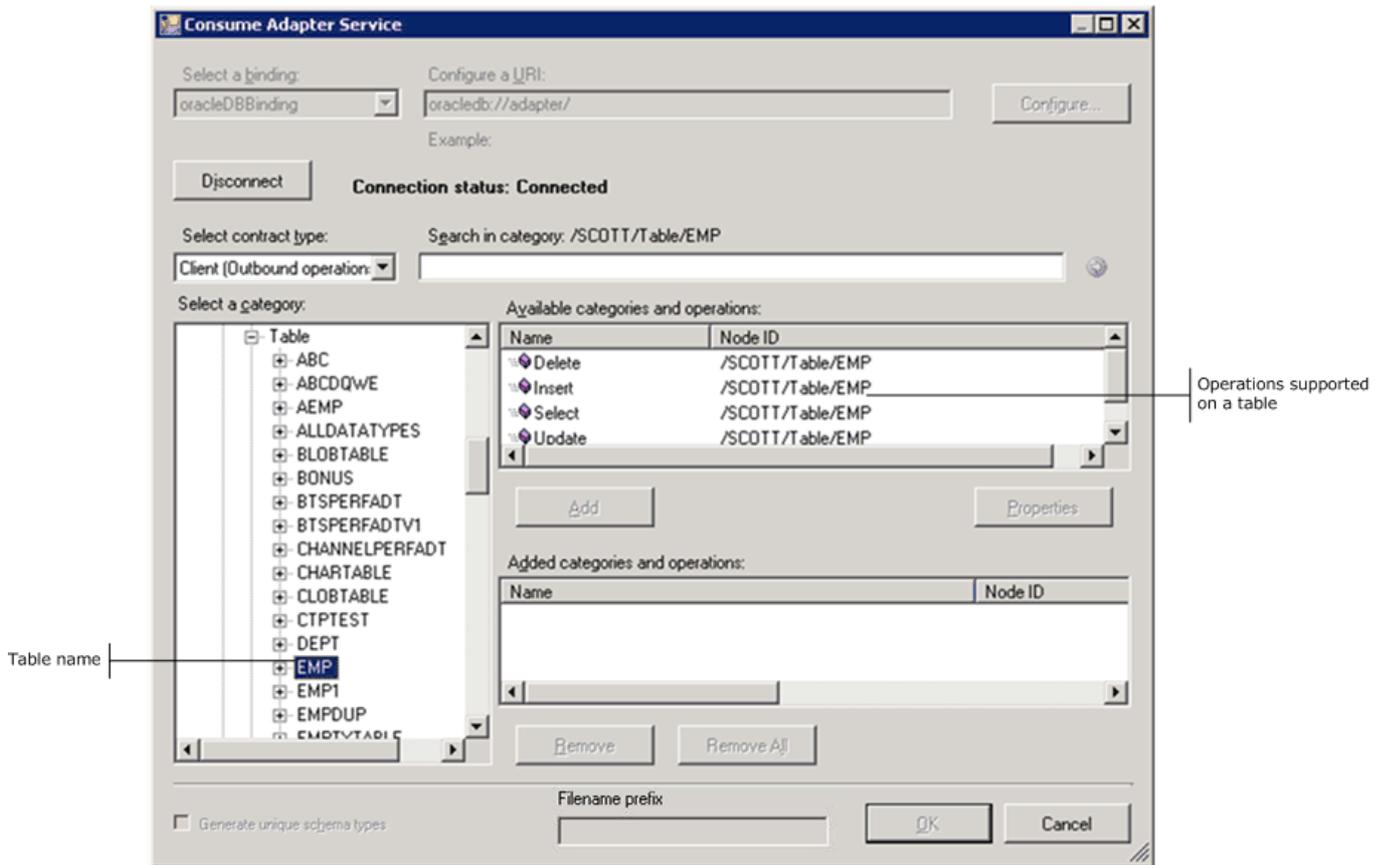
You can directly go to the “immediate” category node or subcategory nodes in the tree, by typing the name of the artifact in while the focus is on the tree view in the **Select a category** box. For example, to jump to the **SCOTT** node, keep the focus on the root node, and then type **SCOTT**.

The following figure shows the Consume Adapter Service Add-in. The SCOTT schema node is selected, and the general category nodes available under the SCOTT node are listed in the **Available categories and operations** box.



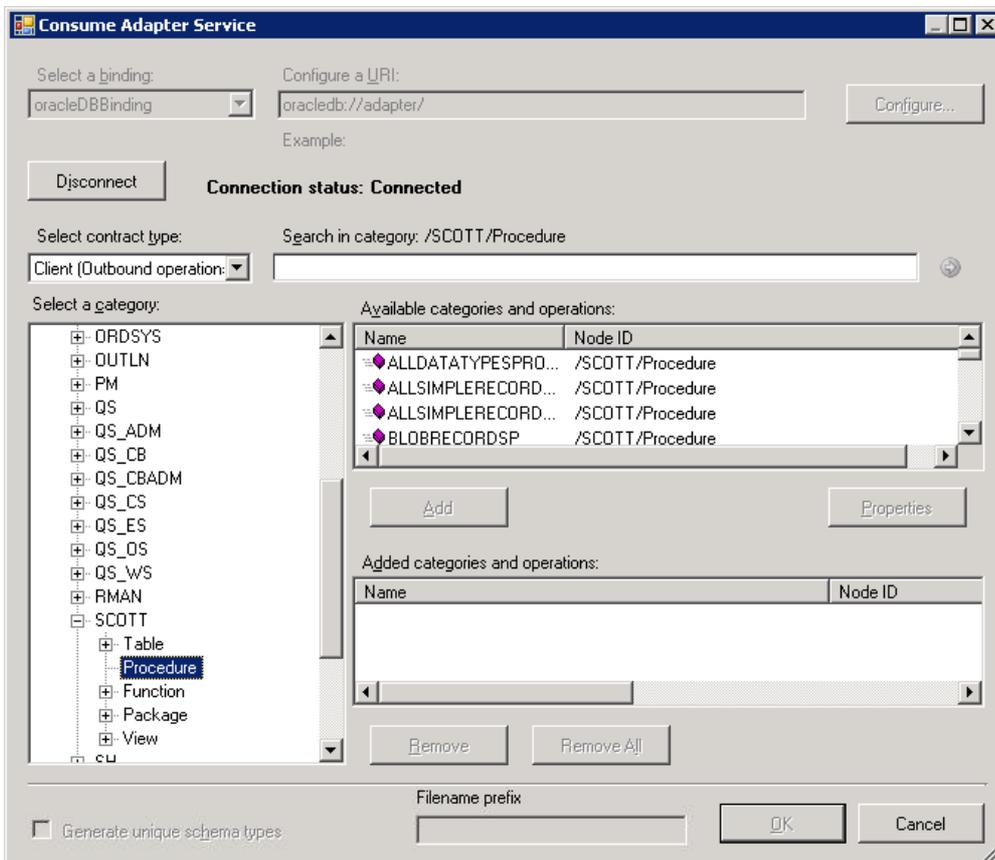
- Click the **Table** node to see the tables for SCOTT in the **Available categories and operations** box. Alternatively, you can see the list of tables by expanding the **Table** node.
- Click a table name to see the operations supported on the table.

The following figure shows the Consume Adapter Service Add-in. The tables available in the SCOTT schema are listed in the **Select a category** box. The operations available for the EMP table are listed in the **Available categories and operations** box.



- Click the **Procedure** node to list the procedures accessible to schema SCOTT in the **Available categories and operations** box.

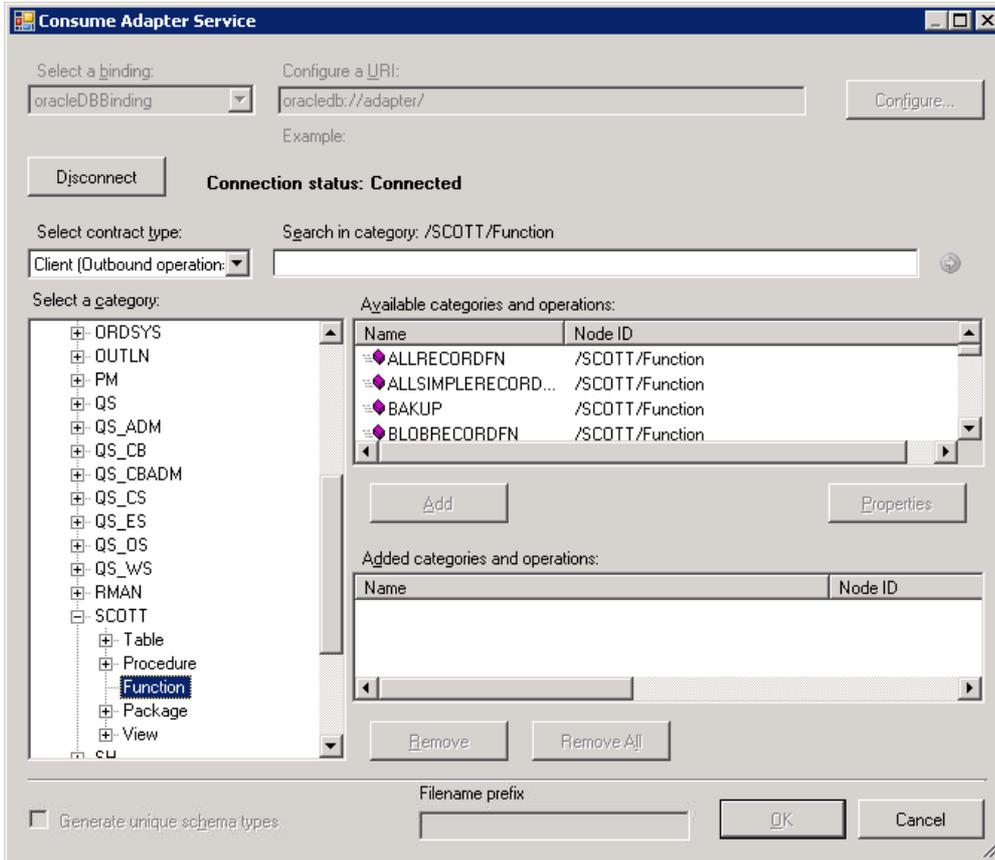
The following figure shows the Consume Adapter Service Add-in. The general category nodes available in the SCOTT schema are listed in the **Select a category** box. The procedures available in the SCOTT schema are listed in the **Available categories and operations** box.



- Click the **Function** node to see the functions for the schema SCOTT in the **Available categories and operations** box.

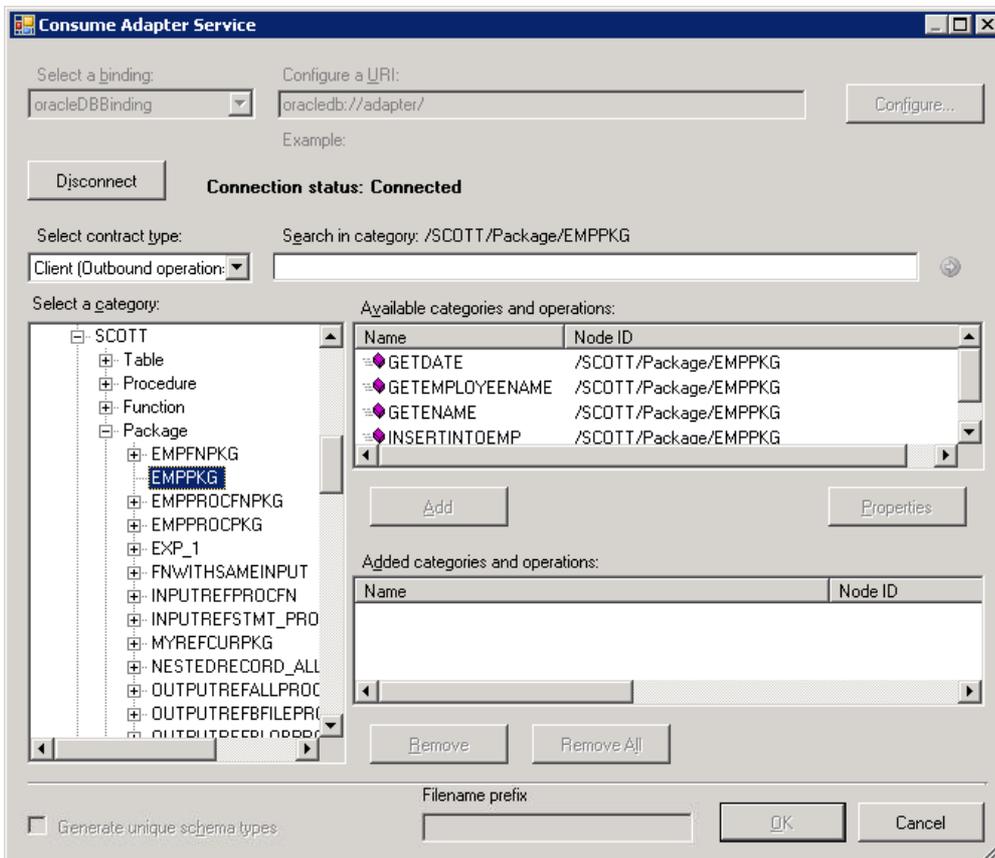
The following figure shows the Consume Adapter Service Add-in. The general category nodes available in the SCOTT schema are listed in the **Select a category** box. The functions available in the SCOTT schema are listed in the **Available categories and**

operations box .



8. Click the **Package** node to see the packages for the schema SCOTT in the **Available categories and operations** box. Alternatively, you can see the list of packages by expanding the **Package** node.
9. Click a package name to see the operations supported on the package.

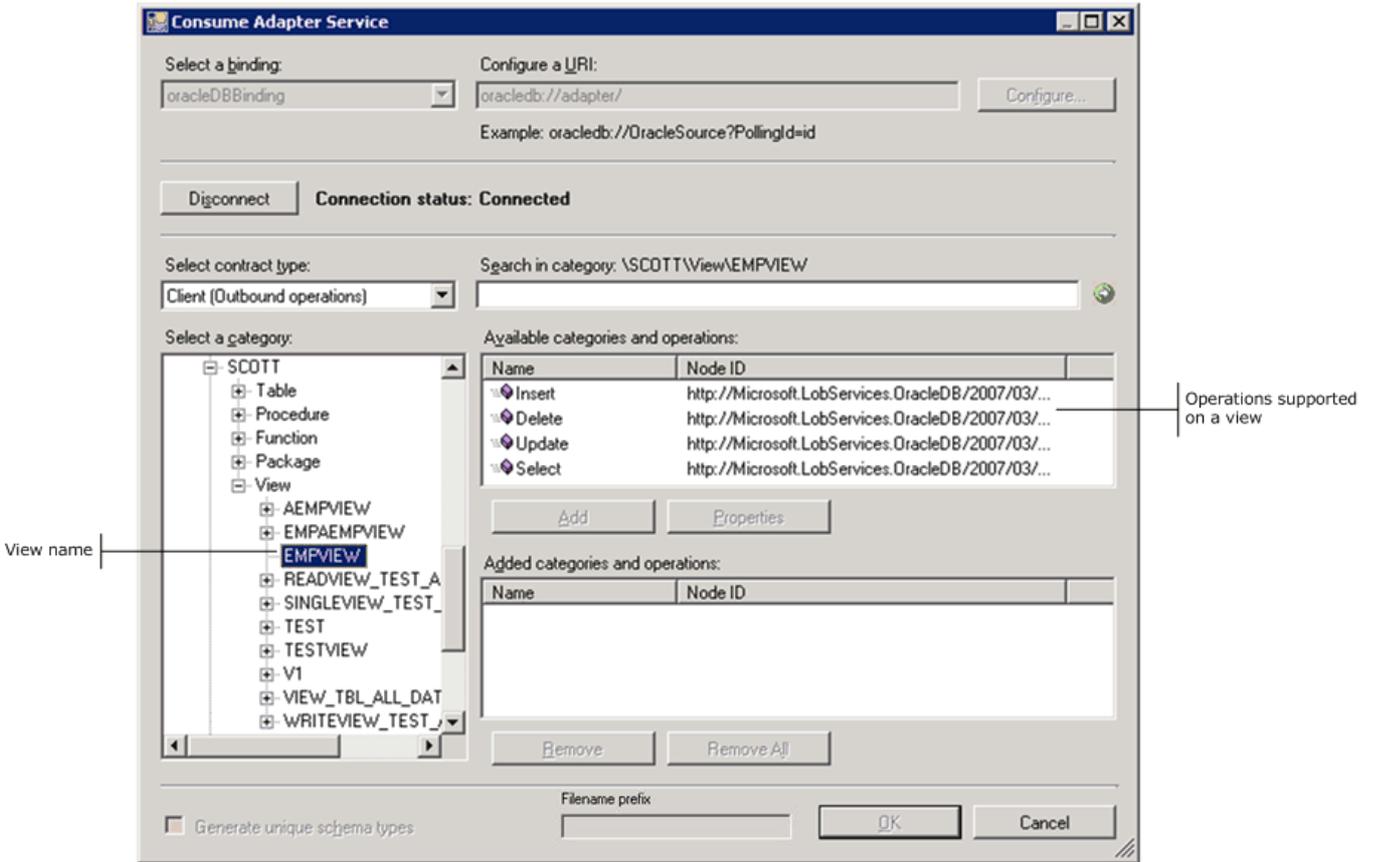
The following figure shows Consume Adapter Service Add-in, which lists the packages, and the supported operations for a particular package, for the SCOTT schema.



10. Click the **View** node to see the views for the schema SCOTT in the **Available categories and operations** box. Alternatively, you can see the list of views by expanding the **View** node.

11. Click a view name to see the operations supported on the view.

The following figure shows the Consume Adapter Service Add-in, which lists the views, and the supported operations for a particular view, for the SCOTT schema.



Note
Using the WCF channel and service models, adapter clients can specify a batch size to perform a batch retrieval of metadata.

Searching for Operations

While searching Oracle metadata using Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in, the Oracle Database adapter:

- Supports wildcard and escape characters in the search expression.
- Enables search immediately under the node at which the search operation is performed. For example, to search for a function, you must be searching under \[Schema]\Functions. Multi-level search is not supported.

The following table lists the special characters that can be used for search and their interpretation by the Oracle Database adapter.

Special character	Interpretation
_ (underscore)	Matches exactly one character For example, A_ matches AB, AC, AD.
% (percentage)	Matches zero or more characters. For example, A% matches A, AB, ABC.
\ (escape)	Escapes the special meaning of % and _ For example, A_B matches A_B.

Note
Escape character is a character that is put before a wildcard character to indicate that the wildcard should be interpreted as a regular character and not as a wildcard.

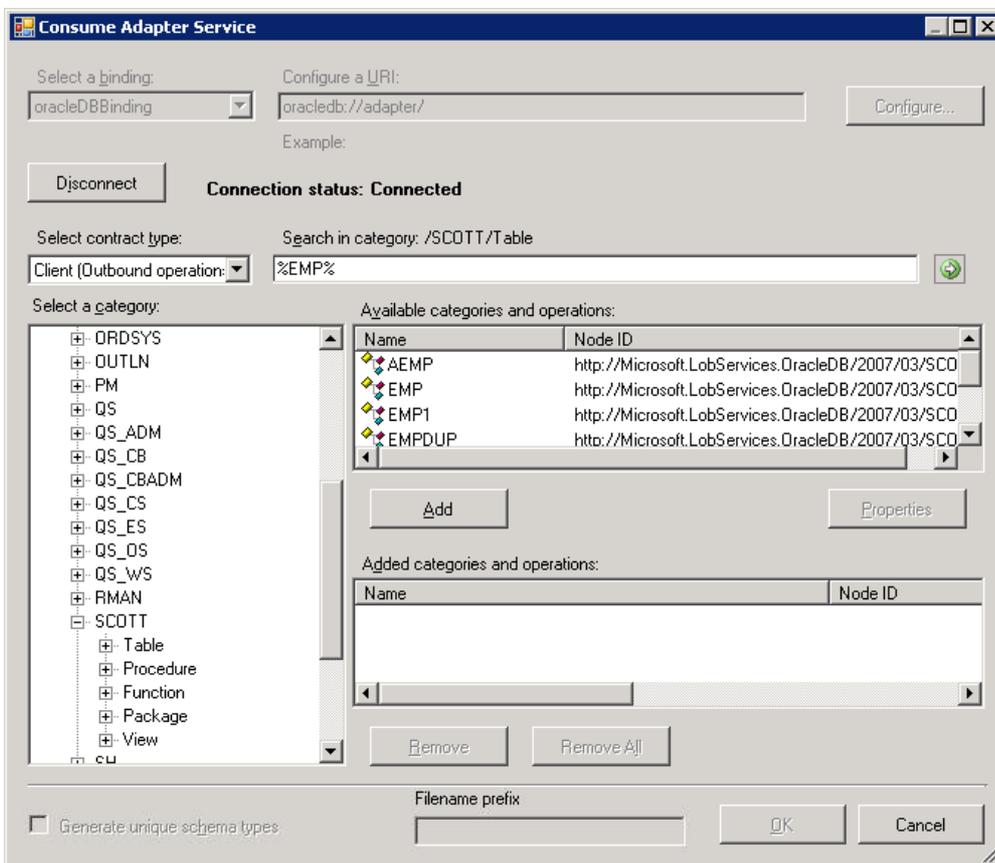
For more information see [How Does the Adapter Surface Oracle Metadata?](#)

To search metadata in an Oracle database

1. Connect to an Oracle database using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in. See [Connecting to the Oracle Database in Visual Studio](#) for instructions.
2. In the Consume Adapter Service Add-in, from the **Select contract type** drop-down list, select the type of contract based on whether you will be searching for inbound or outbound operations using the adapter.
3. In the **Select a category** box, click the schema containing the tables, procedures, functions, packages, and views that you want to search. If you are not sure which schema to click, click the root node.
4. In the **Search in category** text box, enter a search expression to search for a specific schema. For example, to search for schemas that have "SC" in their name, type **%SC%** in the text box.
5. Click the button with the right-arrow icon to start the search. After the search is complete, the **Available categories and operations** box lists the schemas that satisfy the search criteria.
6. In the **Select a category** box, expand the node that corresponds to the schema, and then click the database item that you want to search within. In the **Search in category** text box, enter a search expression to search for a specific database item.

For example, to search for tables that have "EMP" in their names, select **Table**, type **%EMP%** in the **Search in category** text box, and then click the button with the right-arrow icon.

The following figure shows the Consume Adapter Service Add-in, which lists the search result.



Note
Using the WCF channel and service models, adapter clients can specify a batch size to perform a batch-wise search of metadata.

Generating Schema Using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard

You can use the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard to generate schema for selected Oracle database artifacts. Once you have browsed and searched for the artifacts you want to invoke, you can generate schema for those artifacts and send messages, conforming to the schema, to Oracle database. Perform the following steps to retrieve metadata from an Oracle database using the Consume Adapter Service Add-in.

Note
You can select category nodes to return all the operations in that category's sub-tree—for example, you can select an entire table (to generate schema for all the operations in the table) or select specific operations on a table (for example, Insert and Delete) to generate schema for only those operations on a table. For more information about the nodes, see [Metadata Node IDs](#).

To retrieve metadata from an Oracle database

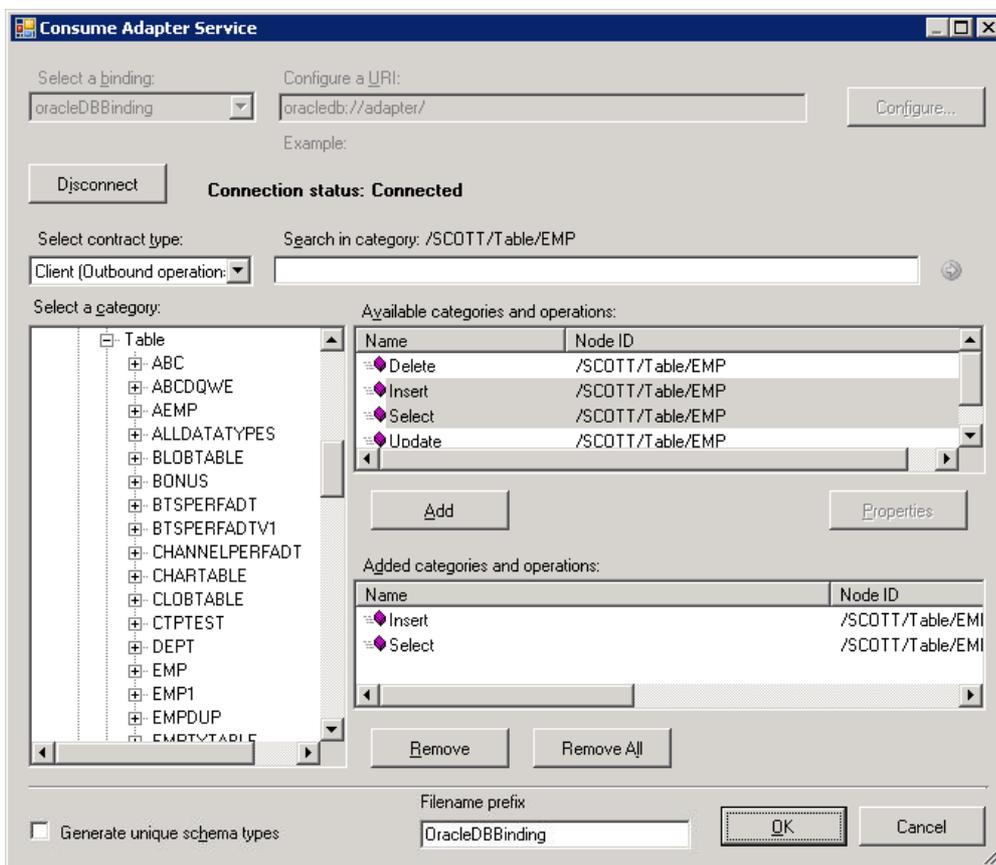
1. Connect to an Oracle database using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard. See [Connecting to the Oracle Database in Visual Studio](#) for instructions.
2. From the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound or outbound operations using the adapter.
3. In the **Select a category** box, expand a schema node.
4. Select the category for which you want to generate metadata. For example, if you want to generate metadata for a table, select **Table**.
5. Expand that particular category node, and select the specific item within that node for which you want to generate metadata.

For example, to generate metadata for a specific table, expand the **Table** node, and select the specific table name.

Note
You can also search for a specific database item, as described in the previous procedure.

6. In the **Available categories and operations** box, select the operations that pertain to the database item that you selected in the previous step, and then click **Add**. The selected operations are listed in the **Added categories and operations** box.

The following figure shows the Consume Adapter Service Add-in, which lists the selected operations.



If you want to generate schema for multiple operations, there may be some duplicate element definitions among these schema that may cause failure in compiling the BizTalk project. For example, consider a scenario where you generate schema for an operation "Op1". The schema for "Op1" contains a parameter of complex data type "CT1". After generating the schema for "Op1" you close the Consume Adapter Service Add-in and re-open it to generate schema for another operation "Op2". Assume that "Op2" also contains a parameter of complex data type "CT1". After you exit the Consume Adapter Service Add-in and compile the project, you will get compilation errors because the complex data type "CT1" is defined twice in different XSD files. In such situations, we recommend the following:

- Generate schema for all the operations in a single run of Consume Adapter Service Add-in. This ensures that the Consume Adapter Service Add-in generates only one definition for the complex data type "CT1".
- If you want to generate schema for multiple operations across different runs of Consume Adapter Service Add-in, make sure you select the **Generate unique schema types** check box so that the generated XSD files contain unique namespaces for the complex data type "CT1".

7. Click **OK**. The schema file is saved with an .xsd extension at the same location as the BizTalk project.

By default, the files are created with the naming convention "OracleDBBindingSchema<n>.xsd", where 'n' can be 1, 2, and so on, depending on the number of schema files created. Alternatively, you can provide a custom name to the schema files by entering a name in the **Filename prefix** text box. The Consume Adapter Service Add-in now creates schema files with the naming convention <file name prefix>Schema<n>.xsd.

Note

The Consume Adapter Service Add-in also creates a binding file (an XML file) containing the binding properties that you specified when generating the schema for an operation and the SOAP action to invoke the operation. You can import this binding file in the BizTalk Server Administration console to create a WCF-Custom port with the connection URI, binding properties, and the SOAP action set. For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Important

Using the Add Adapter Metadata Wizard does not generate a binding file.

8. On the **File** menu, click **Save All**.

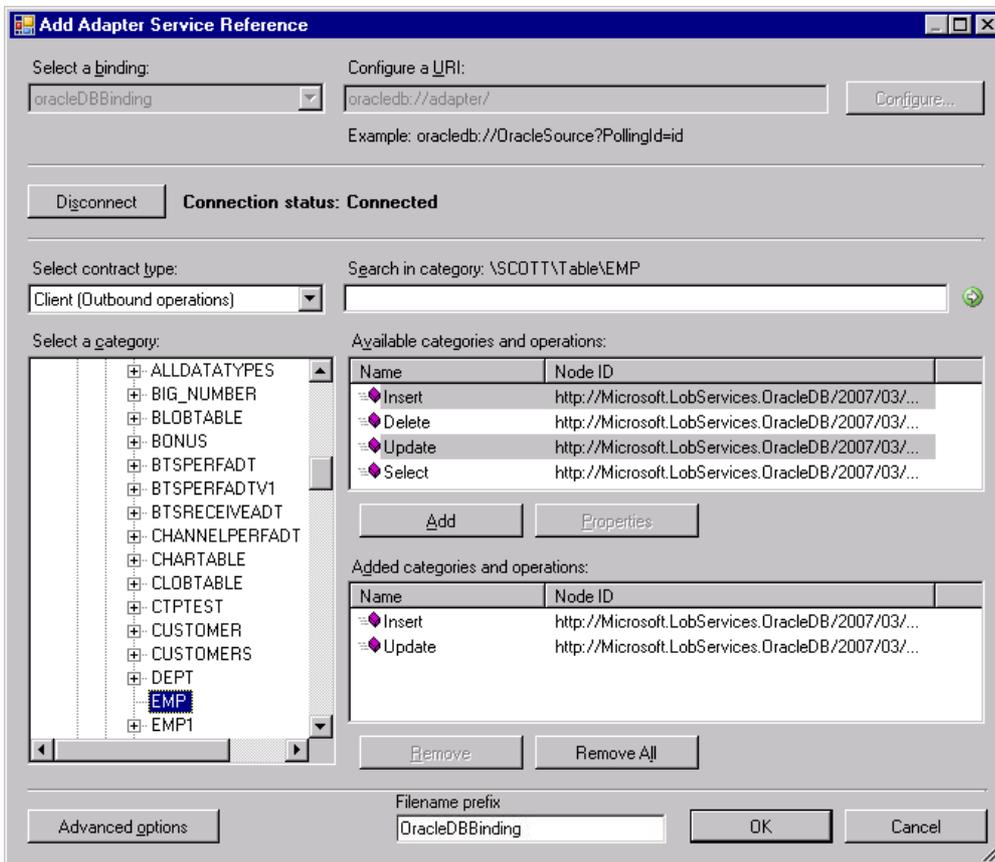
Generating a WCF Client or WCF Service Contract Using the Add Adapter Service Reference Plug-in

You can use the Add Adapter Service Reference Plug-in to generate either WCF client code for outbound operations or WCF service code for the inbound operations.

To retrieve metadata from an Oracle database

1. In the Add Adapter Service Reference Plug-in, from the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound (POLLINGSTMT) or outbound operations.
2. Browse or search for categories (such as an Oracle Database table) or for the specific operations for which you want to generate a WCF client (or WCF service contract).
For example, to browse for the operations in the SCOTT.EMP table, In the **Select a category** box:
 - a. Expand the root node (/) to see the schemas surfaced for the Oracle database.
 - b. Under the root node, expand the **SCOTT** node to see the categories exposed for the SCOTT schema.
 - c. Under the **SCOTT** node, expand the **Table** node to see the tables surfaced for the SCOTT schema.
 - d. Under the **Table** node select the **EMP** node. The operations surfaced for the EMP table are listed in the **Available categories and operations** box.
3. In the **Available categories and operations** box, select the operations or categories for which you want to generate a WCF client (or WCF service contract), and then click **Add**. The selected operations are listed in the **Added categories and operations** box.

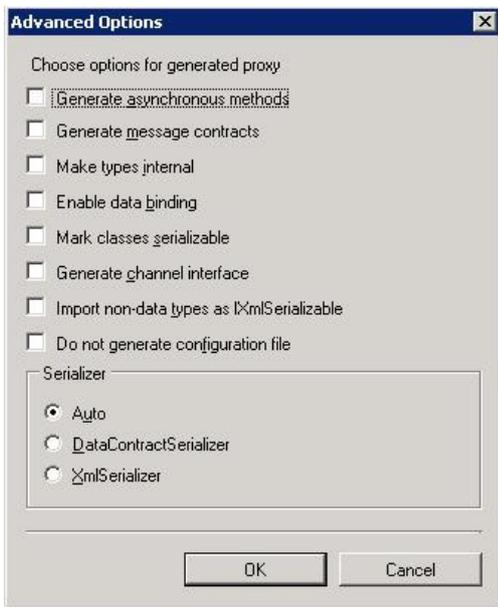
The following figure shows the Add Adapter Service Reference Plug-in with the Insert and Update operations for the SCOTT.EMP table selected.



Important
Depending on the outbound operations (or categories) that you select, more than one WCF client class may be generated. For more details, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

4. For most scenarios the default serialization options are sufficient; however, if needed, you can control several aspects about the code that is generated and the type of serializer that is used. To set these options:
 - a. Click **Advanced Options** to open the **Advanced Options** box.
 - b. In the **Advanced Options** box under **Choose options for generated proxy**, select the options that you want. For example, you can select whether asynchronous methods are generated for the WCF client or disable the generation of a configuration file.
 - c. Under **Serializer** select the serializer that should be used.

The following figure shows the **Advanced Options** box with the default selections (**Auto** is selected for the serializer and no other options are selected).



The options that you can configure in the **Advanced Options** box are equivalent to some of the options available when you use the ServiceModel Metadata Utility Tool (svcutil.exe). For more information about these options, see "ServiceModel Metadata Utility Tool (Svcutil.exe)" at <http://go.microsoft.com/fwlink/?LinkId=99821>.

5. Click **OK**. The Add Adapter Service Reference Plug-in saves the WCF client class (or WCF service interface) and helper code for the operations and categories that you have selected in your project directory. By default, a configuration file is also saved. Slightly different files are generated for inbound and outbound operations; for more information, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

You can select any node that is listed in the **Available categories and operations** box. If you select a category node then all of the operations available under that node and its sub-nodes will be selected. For example, to generate a WCF client for all of the operations surfaced for the EMP table, you can select the EMP node; to generate WCF clients for all of the tables in the SCOTT schema, you can select the Table node; and so on.

See Also

Other Resources

[Retrieving Metadata for Oracle Operations in Visual Studio](#)

Working with BizTalk Adapter for Oracle Database Binding Properties

The Microsoft BizTalk Adapter for Oracle Database surfaces several binding properties. By setting these properties, you can control some of the adapter's behavior. This section describes the Oracle Database adapter binding properties. It also shows how you can access them by using .NET programming or by setting properties on a BizTalk Server physical port binding.

The Adapter Binding Properties

The following table shows the Oracle Database adapter binding properties grouped by category. The category refers to the node under which each binding property appears in the dialog boxes that are presented by different applications to configure the adapter (or binding).

Binding Property	Category	Description	.NET Type
CloseTimeout	General	The WCF connection close timeout. The default is 1 minute. Not supported.	System.TimeSpan
EnableBizTalkCompatibilityMode	General	Set the value of this binding property to True when using the adapter with BizTalk Server. Otherwise, you must set the value of this binding property to False .	boolean (System.Boolean)
InboundOperationType	General	Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling . For more information about Polling see Support for Receiving Polling-based Data-changed Messages . For more information about Notification , see Receiving Database Change Notifications .	enum
Name	General	A read-only value that returns the name of the file generated by the Add Adapter Service Reference Visual Studio Plug-in to hold the WCF client class. The Add Adapter Service Reference Plug-in forms the file name by appending "Client" to the value of the Name property. The value returned is "OracleDBBinding"; for this value, the generated file will be named "OracleDBBindingClient".	string

OpenTimeout	General	<p>ODP.NET property. Specifies the WCF connection open timeout. The default is 1 minute. This property is implemented by using ODP.NET.</p> <p>◆Important The Oracle Database adapter always uses OpenTimeout to set the connection open timeout when it opens a connection to the Oracle database. The adapter ignores any timeout (System.TimeSpan) parameters passed when you open a communication object, such as a channel.</p>	System.TimeSpan
ReceiveTimeout	General	<p>Specifies the WCF message receive timeout. Essentially, this means the maximum amount of time the adapter waits for an inbound message. The default is 10 minutes.</p> <p>◆Important For inbound operations such as polling, we recommend setting the timeout to the maximum possible value, which is 24.20:31:23.6470000 (24 days). When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.</p>	System.TimeSpan
SendTimeout	General	<p>ODP.NET property. Specifies the WCF message send timeout. The default is 1 minute. Not supported.</p>	System.TimeSpan
DataFetchSize	BufferManagement	<p>ODP.NET property. Specifies the amount of data in bytes that ODP.NET fetches from the result set in one server roundtrip. The default is 65536. This property is used for performance tuning.</p>	long (System.Int64)
InsertBatchSize	BufferManagement	<p>Specifies the batch size for multiple record Insert operations. The default is one. For values of InsertBatchSize greater than one, the Oracle Database adapter batches the specified number of records into a single ODP.NET call. If the number of records in the Insert operation is not a multiple of the batch size, the final batch will contain fewer records than the batch size value. For example, if the insert message has 10 records and the InsertBatchSize is set to 1, the adapter reads individual records and writes them into the Oracle database. So, the adapter performs 10 separate operations on the Oracle database. Similarly, if the insert message has 10 records and the InsertBatchSize is set to 5, the adapter will read and write 5 records at a time into the Oracle database, therefore performing only 2 insert operations.</p> <p>If the structure of the records is not the same across a batch, a Microsoft.ServiceModel.Channels.Common.XmlReaderParsingException exception is thrown and the transaction is rolled back for the entire insert operation. A well-chosen value for InsertBatchSize can greatly improve adapter performance for multiple record Insert operations.</p>	int (System.Int32)

Long Data type Column Size	BufferManagement	<p>Specifies the maximum size in bytes (32512) of an Oracle long data type column. The default is 0. You must use the default value if you are not performing operation on long data type. To prefetch the data, you must specify -1 as the value for this binding property. You must explicitly set an appropriate value for this binding property if you are:</p> <ul style="list-style-type: none"> • Executing a stored procedure that contains parameters of long data type. • Performing a Select operation on a table that contains columns with long data type, and the SELECT statement does not include the primary key column. <p> Note This binding property is deprecated.</p>	integer (System.Int64)
Max Output Associative Array Elements	BufferManagement	<p>Specifies the size of the associative array that the adapter creates when performing operations that return an associative array in the response. The adapter communicates the size of the array to ODP.NET, which in turn creates a buffer depending on the array size. Default is 32.</p> <p>This binding property is useful when performing operations involving PL/SQL table types.</p>	integer (System.Int32)
Metadata Pooling	BufferManagement	<p>ODP.NET property. Specifies whether ODP.NET caches metadata information for executed queries. The default is True, which enables metadata pooling. Caching this information improves performance; however, if changes to the underlying Oracle artifacts occur on the Oracle system, this pooled metadata will be out of sync. This might cause operations performed on the Oracle system to return unexpected exceptions. This property is used for performance tuning.</p>	boolean (System.Boolean)
Statement Cache Purge	BufferManagement	<p>ODP.NET property. Specifies whether the ODP.NET statement cache associated with a connection is purged when the connection is returned to the connection pool. The default is False, which disables statement cache purging. This property is used for performance tuning.</p>	boolean (System.Boolean)
Statement Cache Size	BufferManagement	<p>ODP.NET property. Specifies the maximum number of statements that can be cached by each ODP.NET connection. Setting this property to a non-zero value enables statement caching for connections. The default is 10. This property is used for performance tuning.</p>	integer (System.Int32)

EnablePerformanceCounters	Diagnostics	Specifies whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter. The default is False ; performance counters are disabled. The LOB Latency performance counter measures the total time spent by the Oracle Database adapter in making calls to the Oracle database.	boolean
EnableSafeTyping	Metadata	Enables or disables safe typing. The default is False ; safe typing is disabled. This feature controls how the adapter surfaces certain Oracle data types. For more information about safe typing, see Basic Oracle Data Types .	boolean
UseSchemaNamespace	Metadata	<p>Specifies whether the schema name (SCOTT, HR, and so on) is included in the xml namespace for operations and their associated types. The default is True; the schema name is included in the namespace. The advantage of not having scheme name included in the namespace is that if there is a table with same name (for example, EMP) in two different schemas then the same XML can be used to perform the simple SQL operations (Insert, Update, Delete, Select) on both tables.</p> <p>For example, if the UseSchemaNamespace property is true, the namespace for these operations on the SCOTT.EMP table is "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP"; if it is false, the namespace is "http://Microsoft.LobServices.OracleDB/2007/03/Table/EMP".</p> <p>◆ Important The message action is not affected by the UseSchemaNamespace binding property; it always includes the schema name.</p> <p>◆ Important We strongly recommend setting this binding property to True while generating metadata. If you set this property to false, the Oracle schema names (for example, SCOTT) will not be available in the XML namespace of the generated schema. So, if there are two tables with the same name in two different Oracle schemas, and they are added to the same BizTalk project, the BizTalk project will fail to build and deploy. If you want to include such schemas in the same BizTalk project, you must manually edit them to include the Oracle schema name in the XML namespace.</p>	boolean

NotificationPort	Notification	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Default is -1, which signifies that ODP.NET uses a valid, random, unused port number.</p> <p>◆Important Adapter clients will not receive database change notifications if Windows Firewall is turned on. Also, turning off Windows Firewall to receive notifications is not advisable. So, to receive notifications without compromising the security of the client-side computers, we recommend specifying a positive integer value as a port number and then adding that port number to the Windows Firewall exceptions list. If you set this binding property to the default value of -1, ODP.NET uses a random port and adapter clients will not know which port to add to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>⚠Caution If there is more than one application in an application domain receiving notifications using the Oracle Database adapter, the NotificationPort binding property for all applications must be set to the same port number. This is because ODP.NET creates only one listener that listens on one port within an application domain.</p>	int (System.Int32)
NotificationStatement	Notification	<p>Specifies the SELECT statement used to register for getting notifications from Oracle database. An example SELECT statement could resemble the following.</p> <pre style="border: 1px solid black; padding: 5px; margin: 10px 0;">SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌Note You must specify the database object name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p> <p>The adapter gets a notification message from Oracle database only when the result set for the specified SELECT statement changes.</p>	string
NotifyOnListenerStart	Notification	<p>Specifies whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts. Default is True.</p>	bool (System.Boolean)
ConnectionLifetime	OracleConnectionPool	<p>ODP.NET property. Specifies the maximum duration in seconds of a connection. The default is 0. This property is used for performance tuning.</p>	int (System.Int32)

Decr Pool Size	OracleConnectionPool	ODP.NET property. Specifies the number of connections that are closed when an excessive amount of established connections are not in use. The default is 1. This is used for performance tuning.	int (System.Int32)
Incr Pool Size	OracleConnectionPool	ODP.NET property. Specifies the number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool. The default is 5. This property is used for performance tuning.	int (System.Int32)
Max Pool Size	OracleConnectionPool	ODP.NET property. Specifies the maximum number of connections in an ODP.NET connection pool. The default is 100. This property is used for performance tuning. ♦ Important You must set MaxPoolSize judiciously. It is possible to exhaust the number of connections available from ODP.NET, if this value is set too large.	int (System.Int32)
Min Pool Size	OracleConnectionPool	ODP.NET property. Specifies the minimum number of connections in an ODP.NET connection pool. The default is 1. This property is used for performance tuning.	int (System.Int32)
Use OracleConnection Pool	OracleConnectionPool	ODP.NET property. Specifies whether to use the ODP.NET connection pool. The default is True , which enables connection pooling. The Oracle Database adapter implements connection pooling by using the ODP.NET connection pool.	boolean (System.Boolean)

Poll edD ataA vaila bleS tate men t	PollingRecei ve	<p>Specifies the SELECT statement executed to determine whether any data is available for polling for a specific table. The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the value specified for the PollingStatement binding property. If the first cell of the result contains a positive value, the adapter executes the polling statement. For example, a valid statement for this binding property will be:</p> <pre>Select * from <table_name></pre> <p>The default value of this binding property is set to:</p> <pre>SELECT 1 FROM DUAL</pre> <p>This implies that the adapter must continue polling irrespective of whether the table being polled has data or not.</p> <p> Note You must not specify stored procedures for this binding property. Also, this statement must not modify the underlying Oracle database.</p>	str in g
Poll ingA ctio n	PollingRecei ve	<p>Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.</p>	str in g
Poll ingIn terv al	PollingRecei ve	<p>Specifies the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database. The default is 500. The polling interval is used by the adapter for the following:</p> <ul style="list-style-type: none"> • The time interval between successive polls. This interval is used to run the poll and post-poll queries. If these queries are executed within the specified interval, the adapter sleeps for the remaining time in the interval. • The polling transaction timeout value. This value must be set large enough to include the polling statement execution time, the post-poll statement (if specified) execution time, and the time to receive the reply from the client application to commit the transaction. <p>If the client application sends a reply before the polling interval expires, the adapter commits the transaction and waits until the polling interval is reached to execute the next poll.</p> <p>If the client application returns a fault, the adapter terminates the transaction.</p> <p>If the polling interval expires before the client application sends the reply, the transaction will time out. For more information about how to use binding properties in a polling scenario, see Receiving Polling-based Data-changed Messages.</p>	int (S ys te m. Int 32)

PollingStatement	PollingReceive	<p>Specifies the polling statement. You can specify a simple SELECT statement or a stored procedure, function, or a packaged procedure or function for polling.</p> <ul style="list-style-type: none"> • If you want to poll a table or view, you must specify a SELECT query in this binding property. • If you want to poll using a stored procedure, function, or procedure or function within a package, you must specify the entire request message for the respective operation in this binding property. <p>The polling statement is executed only if the statement executed by the PolledDataAvailableStatement binding property returns some data.</p> <p>◆ Important The Oracle Database adapter executes the polling statement and the post-poll statement (if specified) inside of an Oracle transaction. If you are using a SELECT statement in the PollingStatement binding property, we recommend that you specify a FOR UPDATE clause in your SELECT statement. This will ensure that the selected records are locked during the transaction and that the post-poll statement can perform any required updates on the selected records.</p> <p>For more information about how to use binding properties in a polling scenario, including the use of the FOR UPDATE clause; see Receiving Polling-based Data-changed Messages.</p>	string
PollWhileDataFound	PollingReceive	<p>Specifies whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is False.</p> <p>Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for PolledDataAvailableStatement returns that data is available for polling. The adapter then executes the statement specified for the PollingInput binding property. Assuming that the adapter takes just 10 seconds to execute the statement, it will now have to wait for 50 seconds before executing the PolledDataAvailableStatement again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the PollWhileDataFound binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.</p> <p>📌 Note This binding property is applicable both for polling on tables and views and polling using stored procedures, functions, or packaged procedures or functions.</p>	string
PostPollStatement	PollingReceive	<p>Specifies a PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer. The default is null; no post-poll statement is executed. The post-poll statement executes inside the polling transaction. Two common uses for the post-poll statement are to:</p> <ul style="list-style-type: none"> • Update a column in the rows returned in the polling statement to indicate that they have been processed and should be excluded from subsequent polling queries. • Move processed records to a different table. <p>◆ Important If a post-poll statement is specified, PollingInterval should be set large enough for the PL/SQL block to complete before the interval expires.</p> <p>For more information about how to use binding properties in a polling scenario, see Receiving Polling-based Data-changed Messages.</p>	string

SkipNilNodes	Run Time Behavior	<p>Specifies whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as 'nil' in the request XML. This binding property is applicable for inserting or updating records in a table and for RECORD type parameters in stored procedures. Default is True, which means the adapter will skip passing values for nodes that are marked as 'nil'. In this case, the default value in Oracle (if specified) is taken into account for nodes that are marked as 'nil'. If set to False, the adapter explicitly passes a null value for these nodes.</p> <p> Note</p> <ul style="list-style-type: none"> • For nodes that are not present in the request XML, the adapter always skips passing values, irrespective of the value of the SkipNilNodes binding property. • For PL/SQL tables of RECORDS, the adapter always passes a null value for nodes that are either marked as 'nil' or not present in the request XML, irrespective of the value of the SkipNilNodes binding property. <p>The following example explains the difference in the adapter configuration based on the value you set for this binding property. Assume a request XML resembles the following:</p> <pre data-bbox="347 801 667 891"><EMPNO>1000</EMPNO> <ENAME>John</ENAME> <SAL nil='true'></SAL></pre> <p>If SkipNilNodes is set to True, the adapter executes the following command:</p> <pre data-bbox="347 1014 1114 1048">INSERT INTO EMP (EMPNO, ENAME) VALUES (1000, "John");</pre> <p>If SkipNilNodes is set to False, the adapter executes the following query:</p> <pre data-bbox="347 1171 1273 1205">INSERT INTO EMP (EMPNO, ENAME, SAL) VALUES (1000, "John", null);</pre> <p>Note that in the second statement, the adapter explicitly inserts a null value for the parameter "SAL".</p>	bo ol (S er ys te m. Bo ol ea n)
---------------------	-------------------	--	--

Use Ambient Transaction	Transaction	<p>Specifies whether the Oracle Database adapter performs the operations using the transaction context provided by the caller. The default value is True, which means that the adapter always performs the operations in a transaction context, assuming that the client is providing the transactional context. If there are other resources participating in the transaction, the connections created enlist in System.Transaction and are elevated to an MSDTC transaction.</p> <p>However, there can be scenarios where you do not want the adapter to perform operations in a transactional context. For example:</p> <ul style="list-style-type: none"> • While performing a simple SELECT operation on the Oracle database (on a send port). • While specify a polling statement that performs a SELECT operation and does not involve any changes to the table either through a DELETE statement or by invoking a stored procedure (on a receive port). <p>Both these operations do not make any updates to the database table and hence, elevating these operations to use an MSDTC transaction can be a performance overhead. In such scenarios, you can set the binding property to false so that the Oracle Database adapter does not perform the operations in a transaction context.</p> <p> Note Not performing operations in a transactional context is advisable only for operations that do not make changes to the database. For operations that update data in the database, we recommend setting the binding property to true otherwise you might either experience message loss or duplicate messages depending on whether you are performing inbound or outbound operations.</p>	boolean (System.Boolean)
GeneratedUserTypesAssemblyFilePath	UDT .NET Type Generation – Design Time	<p>Specifies the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata. You must specify a DLL name if you are generating metadata for tables or packages, stored procedures, or functions that use UDTs. Specifying the DLL name is optional for tables and views that have UDTs. The generated DLL is saved to the same location as the executable.</p> <p>This binding property is required only while generating metadata.</p> <p> Note</p> <ul style="list-style-type: none"> • You must specify only one filename. For all the UDTs in the metadata, the adapter generates a single file with the given name. If you do not specify a name, the adapter generates the DLL with a GUID name. • This binding property is not available in BizTalk Server while configuring a WCF-OracleDB receive or send port. 	string
GeneratedUserTypesKeyFilePath	UDT .NET Type Generation – Design Time	<p>Specifies the name and path of the key file that the adapter uses to create a strongly-typed assembly.</p> <p>This binding property is optional and is required only while generating metadata.</p> <p> Note This binding property is not available in BizTalk Server while configuring a WCF-OracleDB receive or send port.</p>	string

User Assembly Location Path	UDT .NET Type Generation – Run Time	<p>Specifies the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata. These DLLs are saved at the location you specified for the GeneratedUserTypesAssemblyFilePath binding property while generating metadata. You must manually copy these DLLs to the following locations:</p> <ul style="list-style-type: none"> • For BizTalk projects: Copy the DLLs at the same location as BTSNTSvc.exe. For BizTalk Server 2010, this is available typically under <installation drive>:\Program Files\Microsoft BizTalk Server 2010. • For .NET Projects: Copy the DLLs to the \bin\Development folder within your .NET project folder. <p>This binding property is required only while sending and receiving messages to perform operations on the Oracle database.</p>	string
AcceptCredentialsInUri	Not surfaced by the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in.	<p>Specifies whether the Oracle connection URI can contain user credentials for the Oracle database. The default is False, which disables user credentials in the connection URI. If AcceptCredentialsInUri is False and the Oracle connection URI contains user credentials, the Oracle Database adapter throws an exception. You can set AcceptCredentialsInUri to True if you must specify credentials in the URI. For more information, see The Oracle Database Connection URI.</p>	boolean

How Do I Set Oracle Binding Properties?

You can set the Oracle binding properties when you specify a connection to an Oracle database. For information about how to set binding properties when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to the Oracle Database in Visual Studio Using Consume Adapter Service Add-in](#).

◆ Important

While using the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file (an XML file) or the app.config file respectively. You must manually add the binding property and its value in the binding file or the app.config file, if required.

- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Specifying a Client Binding for the Oracle Database](#).
- Use the WCF ServiceModel Metadata Utility Tool (svcutil.exe), see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Oracle Database](#).

See Also

Concepts

[Development](#)

Streaming and the Oracle Database Adapter

The Microsoft BizTalk Adapter for Oracle Database supports streaming for Oracle large object (LOB) data types. With the Oracle Database adapter operations are invoked and responses are returned by exchanging SOAP messages. A SOAP message body is composed of XML nodes.

There are two kinds of message streaming that are supported by the adapter:

- **Node streaming.** In node streaming each node is buffered by the adapter before it is sent to the Oracle database (or returned to the client). This means that, for an LOB data type, the entire value is read into a buffer.
- **Node-value streaming.** In node-value streaming the actual value of the node can be streamed in chunks between the Oracle database and the client. Node-value streaming supports end-to-end streaming of LOB data types between the adapter client and the Oracle database.

Both of these streaming modes rely on support for node streaming and node-value streaming on messages in WCF. For this reason, streaming for LOB types is tied closely to how messages are created and consumed both by the adapter and by a client application. One result of this is that support for streaming LOB types is not the same across all programming models.

The sections in this topic provide:

- Fundamental background information about how message streaming is supported in WCF and how it is implemented by the adapter.
- Information about how streaming for LOB data types is supported when you use the adapter in each programming model.

Streaming Fundamentals

The support for streaming implemented by the Oracle Database adapter is a combination of:

- Message streaming support in WCF.
- Streaming support in the Oracle client library (ODP.NET).
- The way messages are created and consumed internally by the adapter.

Message Streaming Support in WCF

How WCF supports streaming on a message depends both on how the message is created and how the message is consumed.

- A WCF message is created by using the static **Create** method of **System.ServiceModel.Channels.Message**. This method has several overloads that support different ways of passing the message body. A WCF message can be created by passing the message body using:
 - A **System.Xml.XmlReader**, or
 - A **System.ServiceModel.Channels.BodyWriter**.
- A WCF message can be consumed using
 - An **XmlReader** by calling **Message.GetReaderAtBodyContents()**, or
 - An **XmlDictionaryWriter** by calling **Message.WriteBodyContents(XmlDictionaryWriter)**.

The following table shows how WCF behaves for different combinations of creating and consuming messages.

Message Created With	Message Consumed With	WCF Behavior
XmlBodyWriter	XmlDictionaryWriter	Node-value streaming is supported. WCF pipes the two writers together to enable streaming. Both the XmlBodyWriter and the XmlDictionaryWriter must support node-value streaming for it to occur.
XmlBodyWriter	XmlReader	Node streaming is supported. WCF internally buffers the XmlReader .
XmlReader	XmlDictionaryWriter	Node streaming is supported. WCF internally buffers the XmlReader and calls back into the XmlDictionaryWriter .
XmlReader	XmlReader	Node streaming is supported. WCF internally buffers the XmlReader .

Streaming Support in the Oracle Client Library (ODP.NET)

ODP.NET supports streaming in the following manner:

- Streaming is supported only on Oracle LOB data types.
- For some table (and view) operations, LOB data types are buffered. Therefore, no streaming is supported.

Internal Message Handling by the Adapter

The adapter supports streaming in the following manner:

- The adapter extends **Message** to implement a custom message class, **Microsoft.Adapters.AdapterUtilities.AdapterMessage**. It creates an **AdapterMessage** for all WCF messages that it provides to the adapter client; this includes the response messages for all outbound operations and the request message for the POLLINGSTMT operation. This enables the adapter to support node-value streaming for the ReadLOB operation by providing an **XmlReader** that supports **ReadValueChunk** to adapter clients.
- The adapter consumes all messages received from the client by using a custom implementation of **XmlDictionaryWriter**.
- The adapter creates all messages that it sends to the client by using a custom implementation of **XmlBodyWriter**, except for the ReadLOB response message. (This includes response messages for all outbound operations and the request message for the POLLINGSTMT operation.)

Streaming Support in the WCF Channel Model

The following table provides detailed information about how streaming is supported in the WCF channel model.

Operation	Node-Value Streaming	Description

Table Insert operation	Supported*	Not supported between the adapter and the Oracle database. Supported between the client and adapter.*	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET, and then the insert is performed. However, node-value streaming between the client and the adapter is possible for LOB columns, if the client creates the message with a BodyWriter .
Table Select operation	Supported		The adapter uses a BodyWriter to create the response message. If the client consumes the message using an XmlDictionaryWriter , node-value streaming for LOB columns occurs.
Table Update operation	Supported	Not supported between the adapter and the Oracle database. Supported between the client and adapter.	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET, and then the update is performed. However, node-value streaming between the client and the adapter is possible for LOB columns if the client creates the message with a BodyWriter .
Table Delete operation	Supported	Not supported between the adapter and the Oracle database. Supported between the client and adapter.	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET and then the delete is performed. However, node-value streaming between the client and the adapter is possible for LOB columns if the client creates the message with a BodyWriter .
Table ReadLOB operation	Supported		The ReadLOB operation is primarily designed to stream LOB data columns in the WCF service model. In the WCF channel model, if the client consumes the message using an XmlReader (by invoking the GetReaderAtBodyContents method on the response message), end-to-end node-value streaming occurs. This is because the adapter returns an XmlReader that supports ReadValueChunk calls for the ReadLOB response message. However, it is recommended that you do not use the ReadLOB operation from the WCF channel model. You can use a Select operation or a SQLEXECUTE operation instead.
Table UpdateLOB operation	Supported		The adapter uses an XmlDictionaryWriter to consume the request message. If the client uses a BodyWriter to create the request message, end-to-end node-value streaming for LOB data occurs.
SQLEXECUTE operation	Supported		The adapter uses a BodyWriter to create the response message. If the client uses an XmlDictionaryWriter to consume the response message, end-to-end node-value streaming for the LOB data occurs. End-to-end node-value streaming is not supported for the request message because the adapter must buffer all operands before it can invoke the operation on the Oracle database.

Stored procedure and function operation	Supported	<p>The adapter uses a BodyWriter to create the response message.</p> <p>If the client uses an XmlDictionaryWriter to consume the response message, end-to-end node-value streaming for the LOB data occurs. (This means that streaming is supported for OUT and IN OUT procedure and function parameters in the response message.)</p> <p>End-to-end node-value streaming is not supported for the request message because the adapter must buffer all operands before it can invoke the operation on the Oracle database.</p>
POLLINGSTMT operation	Supported	<p>The adapter uses a BodyWriter to create the POLLINGSTMT request message. If the client consumes the message using an XmlDictionaryWriter, then node-value streaming for LOB columns occurs.</p>

For information about how to implement LOB data streaming in your code when you use the WCF channel model, see [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#).

Streaming Support in the WCF Service Model

Serializing and deserializing between the XML representation of a message and the managed code object representation of that message requires writing and reading the entire message into memory. For this reason, neither node streaming nor node-value streaming is supported for most operations.

The only exception to this is the ReadLOB operation. This operation is implemented specifically to support end-to-end streaming for reading table and view LOB columns in the WCF service model.

Streaming Support in BizTalk Server

The following table provides detailed information about how streaming is supported in BizTalk Server. (All references to the "adapter" refer to the Oracle Database adapter; the WCF-Custom adapter is always referred to by its full name in this table.)

Operation	Node Streaming	Node-Value Streaming	Description
Table Insert operation	Supported*	Not supported between the adapter and the Oracle database; however, data is streamed between BizTalk Server and the adapter.	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET and then the insert is performed. However, node-value streaming between BizTalk Server and the adapter is supported for LOB data types because the WCF-Custom adapter creates the message with a BodyWriter .
Table Select operation	Supported	Supported	The WCF-Custom adapter uses an XmlDictionaryWriter to consume the response message, so end-to-end node-value streaming for LOB types is supported.
Table Update operation	Supported	Not supported between the adapter and the Oracle database; however, data is streamed between BizTalk Server and the adapter.	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET and then the update is performed. However, node-value streaming between BizTalk Server and the adapter is supported for LOB data types because the WCF-Custom adapter creates the message with a BodyWriter .
Table Delete operation	Supported	Not supported between the adapter and the Oracle database; however, data is streamed between BizTalk Server and the adapter.	End-to-end node-value streaming is not supported because the values of LOB columns are buffered by ODP.NET and then the delete is performed. However, node-value streaming between BizTalk Server and the adapter is supported for LOB data types because the WCF-Custom adapter creates the message with a BodyWriter .

Table ReadLOB operation	The ReadLOB operation is not supported for BizTalk Server.	The ReadLOB operation is not supported for BizTalk Server.	The ReadLOB operation is not supported for BizTalk Server. Use the Select operation or a SQLEXECUTE operation instead.
Table UpdateLOB operation	Supported	Supported	The WCF-Custom adapter uses a BodyWriter to create the request message, so end-to-end node-value streaming for LOB data types is supported.
SQLEXECUTE operation	Supported	Supported	The WCF-Custom adapter uses an XmlDictionaryWriter to consume the response message, so end-to-end node-value streaming for LOB data types in the response message is supported. End-to-end node-value streaming is not supported for the request message because the adapter must buffer all operands before it can invoke the operation on the Oracle database.
Stored procedure and function operation	Supported	Supported	The WCF-Custom adapter uses an XmlDictionaryWriter to consume the response message, so end-to-end node-value streaming for LOB data types in the response message is supported. (This means that streaming is supported for OUT and IN OUT procedure and function parameters in the response message.) End-to-end node-value streaming is not supported for the request message because the adapter must buffer all operands before it can invoke the operation on the Oracle database.
POLLINGSTMT operation	Supported	Supported	The WCF-Custom adapter uses an XmlDictionaryWriter to consume the (inbound) request message, so end-to-end node-value streaming for LOB data types is supported.

See Also
Concepts
[Development](#)

Receiving Polling-based Data-changed Messages

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for Oracle Database supports receiving polling-based data-changed messages by polling the Oracle database. The adapter delivers the messages to your application by:

- Executing a SQL SELECT query to determine whether data is available for polling. You can configure the adapter to execute the SQL SELECT query periodically or continuously.
- Executing a SQL SELECT query against an Oracle table or view or executing stored procedures, functions, or packaged procedures and functions.
- Executing an optional post-poll PL/SQL code block on the Oracle database. This code block is often used to update a field on the queried records in the target or to move the queried records to another table or view.
- Returning the query results in a result set by invoking the POLLINGSTMT operation or the stored procedures, functions, or packaged procedures and functions that are exposed as polling operations.

The adapter executes all of these operations inside of an Oracle transaction.

The adapter also enables you to receive data-changes messages for multiple Oracle artifacts in the same application by exposing a *PollingId* parameter in the connection URI. This parameter modifies the target namespace of the POLLINGSTMT operation.

How Do I Modify the Target Namespace of the POLLINGSTMT Operation?

You can modify the target namespace of the POLLINGSTMT operation by setting the *PollingId* query string parameter in the connection URI. If a *PollingId* is specified in the connection URI, the Oracle Database adapter appends the string specified in the *PollingId* parameter to the default target namespace for the POLLINGSTMT operation:

`http://microsoft.lobservices.oracledb/2007/03/POLLINGSTMT`. The message action of the POLLINGSTMT operation is not modified.

For example, if the following connection URI is specified: `OracleDb://User=SCOTT;Password=TIGER@Adapter?`

`PollingId=AcctActivity`, the target namespace will be

`http://microsoft.lobservices.oracledb/2007/03/POLLINGSTMTAcctActivity`.

By providing a unique namespace for each POLLINGSTMT operation, you can receive data-changed messages for multiple Oracle tables and views in your application.

For more information about the Oracle Database adapter connection URI, see [The Oracle Database Connection URI](#).

How Do I Configure the Oracle Database Adapter for Receiving Data-changed Messages Using Binding Properties?

You configure the Oracle Database adapter to receive data-changed messages by setting some or all of the following binding properties.

Bind ing Prop erty	Value	De fa ult	Required/Optional
Inbo und Oper atio nTyp e	Make sure that the value is set to Polling .	Pol lin g	Required. If not explicitly set, the default value will apply.

Poll edD ataA vaila bleS tate men t	<p>Specify the SELECT statement executed to determine whether any data is available for polling for a specific table. The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the value specified for the PollingStatement binding property. If the first cell of the result contains a positive value, the adapter executes the polling statement. For example, a valid statement for this binding property will be:</p> <pre>Select * from <table_name></pre> <p> Note You must not specify stored procedures for this binding property. Also, this statement must not modify the underlying Oracle database.</p>	SE LE CT 1 F RO M DU AL	Required. If not explicitly set, the default value will apply, which implies that the adapter must continue polling irrespective of whether the table being polled has data or not.
Poll ingA ctio n	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consumer Adapter Service Add-in.	nul l	Optional for polling operations on tables and views using the SELECT statement.
Poll ingIn terv al	Set to the interval, in seconds, at which you want the adapter to query the Oracle database. This property specifies the polling interval and the polling transaction time out. The value should be greater than the amount of time it takes to execute the query and post-poll statement (if one is specified) on the Oracle database plus the amount of time it takes for the client to process the query data and return the polling response message.	50 0	Required. If not explicitly set, the default value will apply.
Poll ingSt ate men t	Specify either of the following: <ul style="list-style-type: none"> • SQL SELECT statement that should be executed against the Oracle database. This statement should include a FOR UPDATE clause. For information about the FOR UPDATE clause, see Specifying a FOR UPDATE Clause in the Polling Statement later in this topic. • Request message for a stored procedure, function, or procedure or function within a package that you want to be polled. 	nul l	Required. Setting PollingStatement to a non-null value enables polling.
Poll While Data Found	Specifies whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval	Fal se	Required. If not explicitly set, the default value will apply.
Post Poll Stat eme nt	Set to an optional PL/SQL code block that is executed by the adapter after the query is performed, but before the query data is returned to the client.	nul l	Optional. If no value is specified, a post poll statement is not executed.
<p> Note If you are using the WCF service model or the WCF channel model, you must also set the AcceptCredentialsInUri binding property.</p>			

Specifying a FOR UPDATE Clause in the Polling Statement

If you are using a SELECT statement as the polling statement and executing a post-poll statement that affects the rows specified in the SELECT statement, you must use the FOR UPDATE clause in the polling statement. Specifying a FOR UPDATE clause ensures that the records selected by the polling statement are locked during the transaction and that the post-poll

statement can perform any required updates on them.

⚠Caution

You can have scenarios where in the time window between the polling and post-poll statements, more records are added to the table that meet the condition of the post-poll statement. In such situations, the post-poll statement would update all the records that satisfy the condition and not just the records selected as part of the polling statement.

If a post-poll statement is specified and the polling statement does not contain a FOR UPDATE clause, you will experience one of the following two conditions:

- If **TransactionIsolationLevel** is set to **ReadCommitted**, the post-poll query will not update the selected rows.
- If **TransactionIsolationLevel** is set to **Serializable**, the following target system exception (**Microsoft.ServiceModel.Channels.Common.TargetSystemException**) will occur when the post-poll statement is executed: "ORA-08177 can't serialize access for this transaction". In such a case, you must set the **PollingRetryCount** binding property to define the number of times you want the adapter to retry the same transaction.

For instructions on how to set the transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).

The polling and post-poll statements are executed in a transaction if the adapter clients have configured to use transactions and the value of the **UseAmbientTransaction** binding property is set to **True** in the adapter.

An example of a polling query with the FOR UPDATE option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE
```

Specifying a NOWAIT Clause in the Polling Statement

You may have scenarios where concurrent threads are accessing the table being polled, leading to too many contentions in the table. This may cause the polling query to be blocked to get a lock on table rows. If you are using a SELECT statement as the polling statement, you may want to specify a NOWAIT keyword along with the FOR UPDATE keyword in the SELECT statement. This will cause the polling query execution within the adapter to return immediately if there are locks on rows which the polling query is trying to select. An exception is usually thrown by Oracle under such conditions. Again, adapter clients may use the **PollingInterval** binding property to specify the time interval after which the adapter clients must retry for polling the data.

An example of a polling query with the NOWAIT option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE NOWAIT
```

Specifying a SKIP LOCKED Clause in the Polling Statement

You may have scenarios where due to concurrent threads accessing the table being polled, some rows in the result set of the WHERE clause specified in the polling query are locked. For example, your polling query returns 6 rows from a table; 4 out of these 6 rows are already locked because of some other transaction. In this case, you might want to specify a SKIP LOCKED keyword along with the FOR UPDATE keyword that instructs the database to attempt to lock the rows specified by the WHERE clause, and to skip any rows that are found to be already locked. The unlocked rows in the WHERE clause are locked during the transaction and the post-poll statement can perform any required updates on them so that these rows are not polled again. This ensures that you do not have to wait to receive the polling messages until all the rows specified by the WHERE clause are unlocked.

The SKIP LOCKED keyword is useful in a scenario where you have adapter clients on multiple computers that are polling the same table in a database. You can load balance among the adapter clients by configuring the polling operation in such a way that you receive polling-based data-change messages for the rows specified by the WHERE clause that are unlocked at that point of time, and then update the row to ensure that if a polling-based data-change message is received by an adapter client, the other clients do not get the same message.

An example of a polling query with the SKIP LOCKED option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE SKIP LOCKED
```

Support for Ordered Delivery (FIFO)

In a production environment, polling can be used to monitor the data changes in the Oracle database. These data-changed messages are received by the adapter client using the Oracle Database adapter. Based on business scenarios, it can be critical that the data-changed messages are received by the adapter client in the right order.

The Oracle Database adapter supports ordered delivery or first-in-first-out (FIFO) to maintain the order in which messages are received from the Oracle database. Here are a few considerations related to support for FIFO in inbound scenarios for the Oracle Database adapter.

- If the message is being consumed by an orchestration, the orchestration must have the ordered delivery set for the messages coming from the Oracle Database adapter receive port.
- If the message is being consumed by a send port (in a content-based routing) scenario, the send port must have ordered delivery set for the messages coming from the Oracle Database adapter receive port.

The WCF-Custom or WCF-OracleDB adapter has a property **Suspend request message on failure** that specifies whether to suspend the request message that fails inbound processing. This property can be set on the **Messages** tab of the WCF-Custom or WCF-OracleDB receive port under the **Error handling** section. The following table lists the scenarios describing how the incoming messages are processed based on whether this property is set and the state of the message subscriber (orchestration or port).

Port property	Subscriber in Unenlisted state	Subscriber in Enlisted but Stopped state
Suspend request message on failure property NOT set	<ul style="list-style-type: none"> • Routing Failure Report is generated as a suspended (non-resumable message) • Actual message is not suspended • Post poll query is not executed as transaction gets aborted. Hence polling repeats and fetches the rows again. • Errors reported in the event log to describe what has happened. 	<ul style="list-style-type: none"> • Not considered a "Failure". There are no error messages in the event log. • Actual message is put into the suspended (resumable) queue. • When the subscribing port or orchestration starts, the messages are automatically resumed. If ordered delivery is set on the subscriber, it will be honored. • The messages may also be resumed manually.
Suspend request message on failure property IS set	<ul style="list-style-type: none"> • Routing Failure Report is generated as a suspended (non-resumable message) • Actual message is also suspended • Post poll query is not executed as transaction gets aborted. Hence polling repeats and fetches the rows again. • Errors reported in the event log to describe what has happened. 	<ul style="list-style-type: none"> • Not considered a "Failure". There are no error messages in the event log. • Actual message is put into the suspended (resumable) queue. • When the subscribing port or orchestration starts, the messages are automatically resumed. If ordered delivery is set on the subscriber, it will be honored. • The messages may also be resumed manually.

See Also

Concepts

[Development](#)

[Receiving Polling-based Data-changed Messages by Using the WCF Service Model](#)

[Receiving Polling-based Data-changed Messages by Using the WCF Channel Model](#)

Other Resources

[Polling Oracle Database by Using BizTalk Server](#)

Developing BizTalk Applications

Developing BizTalk applications involves creating a BizTalk project in Visual Studio and using the Consume Adapter Service BizTalk Project Add-in to generate XML schema. Once you have generated the schema, you can either use Content-Based Routing (CBR) or create BizTalk orchestrations to send and receive messages that conform to the generated schema.

CBR can be used in scenarios where the messages being sent to the Oracle database do not require any intensive processing. For example, if you know that the receive port will be receiving messages only of a certain type, you can add a filter to the send port to route the messages matching the filter expression to the send port.

In BizTalk orchestrations, you create send and receive ports to send and receive messages to and from the WCF-Custom adapter, which in turn sends the messages to BizTalk Server. This section provides information about using BizTalk orchestrations to perform operations on the Oracle database using the Oracle Database adapter. The Oracle Database adapter in turn uses the WCF-Custom adapter, which can interact with a WCF binding.

◆ Important

To use the Oracle Database adapter with Microsoft BizTalk Server, you must always set the **EnableBizTalkCompatibilityMode** binding property to **True**. For instructions about how to set the binding properties, see [Specifying Binding Properties](#).

◆ Important

Unlike the earlier version of the adapters shipped with BizTalk Server, the Oracle Database adapter shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console. This is because the latest Oracle Database adapter is a WCF custom binding. So, while the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF custom bindings and hence, does not display the WCF based Oracle Database adapter. Also, to generate metadata you must use the Consume Adapter Service BizTalk Project Add-in and not the Add Adapter Metadata Wizard, as used for the previous version of the adapter. For instructions on using the Consume Adapter Service Add-in, see [Retrieving Metadata for Oracle Operations in Visual Studio](#). For more such differences between the adapter versions, see [Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#).

In This Section

- [Before You Develop BizTalk Applications](#)
- [Building Blocks for Developing BizTalk Applications](#)
- [Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server](#)
- [Performing Operations on Tables with Large Object Types Data by Using BizTalk Server](#)
- [Invoking Functions and Procedures by Using BizTalk Server](#)
- [Invoking Overloaded Functions and Procedures by Using BizTalk Server](#)
- [Invoking Functions and Procedures with REF CURSORS by Using BizTalk Server](#)
- [Invoking Functions and Procedures with RECORD Types by Using BizTalk Server](#)
- [Performing Operations on Tables with BFILE Data Types by Using BizTalk Server](#)
- [Performing a SQLEXECUTE Operation by Using BizTalk Server](#)
- [Polling Oracle Database by Using BizTalk Server](#)
- [Performing Composite Operations on Oracle Database by Using BizTalk Server](#)
- [Receiving Database Change Notifications Using BizTalk Server](#)

See Also

Concepts

[Development](#)

Before You Develop BizTalk Applications

This section provides information about what you must do before developing BizTalk applications using the Oracle Database adapter. The section also lists some BizTalk Server tools that are used to develop BizTalk applications.

In This Section

- [Create a Strong-Name Key File](#)
- [Learn About BizTalk Server Tools](#)

Create a Strong-Name Key File

You must create a strong-name key file to build projects in Microsoft Visual Studio. A strong name consists of the project's identity—its simple text name, version number, and culture information (if provided)—plus a public key and a digital signature. The strong-name key file contains the public key and the private key.

Important

Creating a strong-name key file is a one-time task. You can use the same key for all the BizTalk applications you develop.

To create a strong-name key file

1. Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2010**, point to **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
2. At the command prompt navigate to the location where you want to create the key file. For example, type **cd C:\Sample**, and then press ENTER.
3. At the command prompt, type **sn -k <key file name>.snk**, and then press ENTER.

Note

You should receive a message at the command prompt stating that the key pair was written to the strong-name key file.

4. At the command prompt, type **exit**, and then press ENTER.

See Also

Concepts

[Learn About BizTalk Server Tools](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Learn About BizTalk Server Tools

The topics on how to use the Oracle Database adapter in [Developing BizTalk Applications](#) assume that you have working knowledge of a number of BizTalk Server tools. You will use the following tools to develop BizTalk applications using Oracle Database adapter:

- Microsoft Visual Studio 2010.
- BizTalk Explorer
- Orchestration Designer
- Pipeline Designer
- BizTalk Mapper
- BizTalk Server Administration console

Prerequisites

You must install Microsoft BizTalk Server 2010 before you can access the BizTalk Server tools.

BizTalk Server Tools

The following table includes topics in the BizTalk Server documentation that explain how to use each of the listed tools.

Tool	Topics in BizTalk Server 2010 Documentation
Visual Studio	<ul style="list-style-type: none"> • Using Visual Studio (http://go.microsoft.com/fwlink/?LinkId=107180) • Working with BizTalk Projects (http://go.microsoft.com/fwlink/?LinkId=107181) • Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=107182) <p>Learn more about Visual Studio:</p> <ul style="list-style-type: none"> • Introduction to Solutions, Projects, and Items (http://go.microsoft.com/fwlink/?LinkID=196962) • Solution Explorer Concepts (http://go.microsoft.com/fwlink/?LinkID=196963)
Orchestration Designer	<ul style="list-style-type: none"> • Creating Orchestrations Using Orchestration Designer (http://go.microsoft.com/fwlink/?LinkId=107184)
Pipeline Designer	<ul style="list-style-type: none"> • Creating Pipelines Using Pipeline Designer (http://go.microsoft.com/fwlink/?LinkId=107185)
BizTalk Mapper	<ul style="list-style-type: none"> • Creating Maps Using BizTalk Mapper (http://go.microsoft.com/fwlink/?LinkId=107186)
BizTalk Server Administration console	<ul style="list-style-type: none"> • Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkId=107187)

See Also

Tasks

[Create a Strong-Name Key File](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Building Blocks for Developing BizTalk Applications

Performing operations on an Oracle database by using the Oracle Database adapter involves two sets of tasks: design-time and run-time.

Design-time Tasks

The WCF LOB Adapter SDK provides capability to browse, search, and retrieve the Oracle metadata for tables, stored procedures, and other such items in the form of XML Schema definition languages (XSDs) by using the Consume Adapter Service BizTalk Project Add-in. The XSDs are specific to the operation you want to perform on the Oracle database. The Consume Adapter Service Add-in is available only when you create a BizTalk project. At design time you need to perform the following tasks:

- **Create BizTalk project and generate schema.** You must create a BizTalk project in Microsoft Visual Studio and generate the schema for the operation that will be performed on the Oracle database. For example, if you want to insert a record into the EMPLOYEE table, you must generate the metadata for the Insert operation for the EMPLOYEE table. In this step, you use the Consume Adapter Service Add-in to generate the schema. For more information, see [Retrieving Metadata for Oracle Operations in Visual Studio](#).
- **Set up an orchestration.** Once you have generated the schema, you must set up an orchestration by using the Orchestration Designer. For a basic orchestration, you add the Send and Receive shapes along with the Send and Receive logical ports. In later steps, you map these logical ports to physical ports by using the BizTalk Server Administration console. The orchestration uses these ports to pick messages that an adapter client sends. The orchestration then passes the messages to the Oracle database. Once a response is received from the Oracle database, the orchestration passes the response to the adapter client.
- **Create messages and link to schema.** In your orchestration, you must create messages that will be mapped to the schema you generated in the first step. Typically, you create a request message and a response message. These messages are mapped to corresponding request and response schemas.
- **Map message shapes to messages and ports.** In your orchestration, you must now map each shape that you added in the second step to messages that you created in the third step. You must also map a message shape to the port on which that message will be sent.

For example, if the first shape in your orchestration is a Receive shape that will receive a message, you map this shape to a request message and the port that sends the request message.

- **Build and deploy the BizTalk project.** After you have set up the orchestration and mapped messages, ports, and schemas, you must build the BizTalk solution. For building a project in Visual Studio, you need an assembly key file. After you successfully build the solution, you must deploy the solution.

Note

More detailed description of these high-level tasks, including procedural information, is provided in various topics of this section.

Once the solution is deployed, your design-time tasks are accomplished. You must now perform the run-time tasks.

Run-time Tasks

At run time, you can use the BizTalk Server Administration console to deploy and monitor the orchestration you created at design time. In addition, you must:

- **Configure the application.** The BizTalk project you deployed at design time shows up in the BizTalk Server Administration console as an orchestration. You must configure this orchestration by mapping the logical ports you created at design time to physical ports that you must now create using the BizTalk Server Administration console.

On the physical ports, you must specify an "action" or "action mapping". This action corresponds to the operation you want to perform on the Oracle database. You need to set the action if you are not using dynamic actions. For more

information about dynamic actions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

- **Start the application.** After the application is configured, you must start the application, and drop input messages at a defined file location. The orchestration consumes the input messages and passes them to the Oracle database and receives a response. This response will be available to you at another defined file location.

To accomplish these high-level design-time and run-time tasks, you must also perform other tasks. For example, when you use the Consume Adapter Service BizTalk Project Add-in to generate the schema, you must specify a connection URI to connect to the Oracle database. This section provides information on such repetitive tasks that you must perform as you develop BizTalk applications using the Oracle Database adapter.

In This Section

- [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#)
- [Specifying the Connection URI](#)
- [Specifying Credentials for the Oracle Database](#)
- [Specifying Binding Properties](#)
- [Specifying SOAP Action](#)
- [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#)
- [Configuring a Physical Port Binding by Using a Port Binding File](#)
- [Configuring Dynamic Ports](#)
- [Reusing Adapter Bindings](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

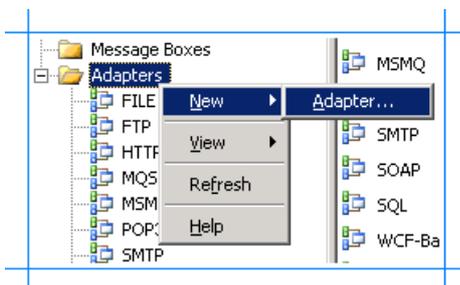
Adding the Oracle Database Adapter to BizTalk Server Administration Console

This topic provides instructions on how to add the WCF-OracleDB adapter to the BizTalk Server Administration console.

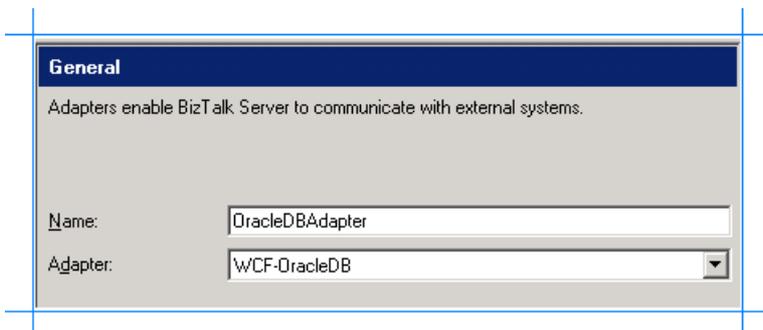
Important
You need not perform these tasks if you want to configure a WCF-Custom port for the Oracle Database adapter.

To add the Oracle Database Adapter

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, expand **Platform Settings**, and then click **Adapters**.
3. Right-click **Adapters**, point to **New**, and click **Adapter**.



4. In the **Adapter Properties** dialog box, specify a name for the adapter and from the **Adapter** list, select **WCF-OracleDB**.



5. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying the Connection URI

A connection URI is a connection string that contains parameters required to connect to the Oracle database. While using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard in Visual Studio, you must specify the URI to connect to the Oracle database to generate the metadata. While configuring an orchestration using the BizTalk Server Administration console, you must specify the URI to connect to the Oracle database to perform operations.

Specifying the Connection URI from Visual Studio

From Visual Studio, you must specify the credentials using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify the Connection URI using Consume Adapter Service Add-in

1. Right-click your BizTalk project, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleDBBinding**, and click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle database:
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type **/** in the **User name** text box and leave the **Password** text box blank.
6. Click the **URI Properties** tab, and specify values for different parameters. For more information about the connection URI for the Oracle Database adapter, see [The Oracle Database Connection URI](#).
7. Click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
8. Click **OK**.

To specify the Connection URI using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleDB**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

Important

If you already have a WCF-OracleDB port configured in BizTalk, select the port from the **Port** list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleDBBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle database:
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
8. Click the **URI Properties** tab, and specify values for different parameters. For more information about the connection URI for the Oracle Database adapter, see [The Oracle Database Connection URI](#).
9. Click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
10. Click **OK**.

Specifying the Connection URI from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the credentials as part of the WCF-Custom or WCF-OracleDB port configuration.

To specify the Connection URI for the WCF-Custom Port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. For a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
5. For a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and

Password text boxes.

- To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
- Select **Get credentials from affiliate application** option, and specify an affiliate application.

6. Click **OK**.

To specify the Connection URI for the WCF-OracleDB port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select **WCF-OracleDB**, and then click **Configure**.

 **Note**

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the port properties dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **oracleDBBinding**.
6. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to the Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
7. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to the Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

8. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Concepts

[Connecting to the Oracle Database Using Windows Authentication](#)

Specifying Credentials for the Oracle Database

The Oracle Database adapter requires the adapter clients to provide client credentials. The adapter uses these credentials to authenticate the user with the Oracle database and to establish a connection.

Adapter clients can provide the client credentials both when using Visual Studio and when using the BizTalk Server Administration console. When using Visual Studio, credentials are required to generate the metadata. When using the BizTalk Server Administration console, credentials are required to perform operations on the Oracle database. This topic provides information about specifying client credentials in Visual Studio and the BizTalk Server Administration console.

Specifying Client Credentials from Visual Studio

From Visual Studio, you must specify the credentials using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify credentials using Consume Adapter Service Add-in

1. Right-click your BizTalk project, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleDBBinding**, and click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle database:
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type **/** in the **User name** text box and leave the **Password** text box blank.

6. Click **OK**.

To specify credentials using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleDB**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important

If you already have a WCF-OracleDB port configured in BizTalk, select the port from the **Port** list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleDBBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle database:
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type **/** in the **User name** text box and leave the **Password** text box blank.
8. Click **OK**.

Specifying Client Credentials from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the credentials as part of the WCF-Custom or WCF-OracleDB port configuration.

To specify client credentials for the WCF-Custom Port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. For a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type **/** in the **User name** text box and leave the **Password** text box blank.
 - Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
5. For a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type **/** in the **User name** text box and leave the **Password** text box blank.
 - Select **Get credentials from affiliate application** option, and specify an affiliate application.
6. Click **OK**.

To specify credentials for the WCF-OracleDB port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select **WCF-OracleDB**, and then click **Configure**.

 **Note**

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the port properties dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **oracleDBBinding**.
6. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to the Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
7. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to the Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.
8. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Concepts

[Connecting to the Oracle Database Using Windows Authentication](#)

Specifying Binding Properties

The Microsoft BizTalk Adapter for Oracle Database surfaces several binding properties that enable you to control some of its behavioral characteristics. This section provides information about setting the binding properties from Visual Studio and from the BizTalk Server Administration console. From Visual Studio, you must specify the binding properties while generating schema for specific operations. From BizTalk Server, you must specify the binding properties as part of the send or receive port for sending or receiving messages from the Oracle database.

For information about the binding properties, including a list of binding properties for Oracle Database adapter, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Specifying Binding Properties from Visual Studio

From Visual Studio, you must specify the credentials using the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Metadata Wizard.

To specify binding properties using Consume Adapter Service Add-in

1. Right-click your BizTalk project and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **oracleDBBinding**, and click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Binding Properties** tab and specify the different binding properties.
6. Click **OK**.

To specify binding properties using Add Adapter Metadata Wizard

1. Right-click your BizTalk project and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleDB**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-OracleDB port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **oracleDBBinding**, and click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the schema. For more information about binding properties, see

Note

If you selected an existing WCF-OracleDB send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such a case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

8. Click **OK**.

Specifying Binding Properties from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the binding properties as part of the WCF-Custom or WCF-OracleDB port configuration.

To specify binding properties for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

4. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab.
5. From the **Binding Type** drop-down list, select **oracleDBBinding**.
6. In the **Configuration** box, specify the values for the different binding properties and click **OK**.

To specify binding properties for the WCF-OracleDB port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-OracleDB adapter you added earlier, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the transport properties dialog box, click the **Binding** tab, and specify values for binding properties.

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying SOAP Action

To perform any operation on the Oracle database using the WCF-based Oracle Database adapter, adapter users must specify a SOAP action. The SOAP action communicates to the adapter what action should be performed. You can specify the SOAP action either at design time or at run time. However, if you specify the SOAP action both at design time and run time, the action you specified at design time will be overridden.

For more information about specifying SOAP action, see <http://go.microsoft.com/fwlink/?LinkId=102063>.

Specifying SOAP Action from Visual Studio

From Visual Studio, you must specify the SOAP action as part of the orchestration by using an **Expression** shape.

To specify SOAP action from Visual Studio

1. In the BizTalk orchestration, include an **Expression** shape by dragging it from the **BizTalk Orchestration** toolbox.
2. Double-click the **Expression** shape to open the BizTalk Expression Editor.
3. Specify the action in the BizTalk Expression Editor. For example:

```
OutboundMessage(WCF.Action)="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert"
```

For more information about **Expression** shape and the BizTalk Expression Editor, see <http://go.microsoft.com/fwlink/?LinkId=102064>.

Specifying SOAP Action from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the SOAP action as part of the WCF-Custom or WCF-OracleDB port configuration.

To specify a SOAP action for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:
 - **By using the single action format.** Use this format if the WCF-Custom port sends and receive messages for a single operation. For example:

```
http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert
```

- **By using the action mapping format.** Use this format if a single WCF-Custom port sends and receives messages for more than one operation. For example, if a single WCF-Custom port sends and receives messages for Op1 (to insert records in the EMP table) and Op2 (to update records in the EMP table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/
Table/EMP/Insert" />
  <Operation Name="Op2" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/
Table/EMP/Update " />
</BtsActionMapping>
```

This approach provides greater flexibility in terms of specifying a set of actions and hence enabling messages belonging to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about action format for each operation, see individual topics under [Technical Reference](#).

To specify a SOAP action for the WCF-OracleDB port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-OracleDB port you added earlier, and then click **Configure**.
5. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
6. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:
 - **By using the single action format.** Use this format if the WCF-OracleDB port sends and receive messages for a single operation. For example:

```
http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert
```

- **By using the action mapping format.** Use this format if a single WCF-OracleDB port sends and receives messages for more than one operation. For example, if a single WCF-OracleDB port sends and receives messages for Op1 (to insert records in the EMP table) and Op2 (to update records in the EMP table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>  
  <Operation Name="Op1" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert" />  
  <Operation Name="Op2" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Update " />  
</BtsActionMapping>
```

This approach provides greater flexibility in terms of specifying a set of actions and hence enabling messages belonging to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about action format for each operation, see individual topics under [Technical Reference](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Manually Configuring a Physical Port Binding to the Oracle Database Adapter

This section provides information about configuring the Microsoft BizTalk Adapter for Oracle Database as a WCF-Custom binding or WCF-OracleDB binding by using the BizTalk Server Administration console. After deploying the adapter, you will be able to send and receive messages from the Oracle database by using the BizTalk Server Administration console. The steps for deploying the adapter vary depending on:

- The direction of communication between BizTalk Server and Oracle Database adapter. You may choose to configure a send, receive, send-receive, or a receive-send port. Your choices are summarized in the following table.

Port direction	Communication pattern	Direction of communication to choose from
Send	One-way	I will always be sending messages on this port.
Receive	One-way	I will always be receiving messages on this port.
Send-receive	Request-response	I will be sending a request and receiving a response.
Receive-send	Solicit-response	I will be receiving a request and sending a response.

For more information, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

- Whether the adapter sends messages to the Oracle database or receives messages from the Oracle database. Depending on whether you want to send or receive messages, you will create a send or receive port, respectively.

 Note
You can also configure the send or receive ports by importing a binding configuration file that is created by the Consumer Adapter Service Add-in as part of metadata generation. For instructions on configuring ports using this binding file, see Configuring a Physical Port Binding by Using a Port Binding File .

In This Section

- [Configuring a Port Using the WCF-Custom Adapter](#)
- [Configuring a Port Using the WCF-OracleDB Adapter](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Configuring a Port Using the WCF-Custom Adapter

This topic provides instructions on how to configure WCF-Custom send and receive ports to perform outbound and inbound operations on the Oracle database using the Microsoft BizTalk Adapter for Oracle Database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To deploy adapters for sending messages to an Oracle database

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the Oracle Database adapter.
4. Right-click **Send Ports**, point to **New**, and point to a type of port you want to configure depending on the mode of communication between BizTalk Server and the Oracle database.
5. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
6. From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
7. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for the Oracle database. For more information about the connection URI, see [The Oracle Database Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on an EMPLOYEE table under the HR schema in an Oracle database is:


```
http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEE/Select
```
 - c. Click the **Binding** tab, and from the **Binding Type** drop-down list, select **oracleDBBinding**. You can specify the different binding properties exposed by the Oracle Database adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
 - d. Click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select the **Use Single Sign-On** option, and specify an affiliate SSO application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

To return to the **Send Port Properties** dialog box, click **OK**.

8. From the **Send handler** drop-down list, select **BizTalkServerApplication**.
9. If you chose **Static One-Way Send Port** in step 4, specify a send pipeline. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
10. If you chose **Static Solicit-Response Port** in step 4, specify send and receive pipelines.
 - a. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - b. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
11. Click **OK**.

To deploy adapters for receiving messages from an Oracle database

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the Oracle Database adapter.
4. Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and the Oracle database.
5. In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
6. On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
7. In the **Receive Location Properties** dialog box, do the following:
 - a. Specify a name for the receive location.
 - b. From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
8. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for the Oracle database. For more information about the connection URI, see [The Oracle Database Connection URI](#).
 - b. Click the **Binding** tab, and from the **Binding Type** drop-down list, select **oracleDBBinding**. You can specify the different binding properties exposed by the Oracle Database adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
 - c. Click the **Others** tab, and do one of the following:
 - Select **User account**, and specify the user name and password to connect to an Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
 - Select **Get credentials from affiliate application** option, and specify an affiliate application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

To return to the **Receive Location Properties** dialog box, click **OK**.

9. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.

10. If you chose **One-way Receive Port** in step 4, specify a receive pipeline. From the **Receive pipeline** drop-down list, select the pipeline corresponding to XMLReceive.
11. If you chose **Request Response Receive Port** in step 4, specify send and receive pipelines.
 - a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
12. In the **Receive Location Properties** dialog box, click **OK**.
13. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Tasks

[Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#)

Concepts

[Connecting to the Oracle Database Using Windows Authentication](#)

Configuring a Port Using the WCF-OracleDB Adapter

This topic provides instructions on how to configure WCF-OracleDB send and receive ports to perform outbound and inbound operations on the Oracle database using the Microsoft BizTalk Adapter for Oracle Database.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To deploy adapters for sending messages to Oracle Database

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
4. Expand the application under which you want to deploy the Oracle Database adapter.
5. Right-click **Send Ports**, point to **New**, and then point to the type of port you want to configure depending on the mode of communication between BizTalk Server and the Oracle database.
6. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
7. From the **Type** drop-down list, select WCF-OracleDB, and then click **Configure**.
8. In the transport properties dialog box, do the following:
 - a. Click the **General** tab, click the **Configure** button and provide values for the connection parameters. For more information about the connection URI, see [The Oracle Database Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on an EMPLOYEE table under the HR schema in an Oracle database is:

```
http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEE/Select
```

- c. Click the **Binding** tab and specify values for the binding properties exposed by the Oracle Database adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to the Oracle database.
 - To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

e. To return to the **Send Port Properties** dialog box, click **OK**.

9. From the **Send handler** list, select **BizTalkServerApplication**.
10. If you chose **Static One-Way Send Port** in step 5, specify a send pipeline. From the **Send pipeline** list, select the pipeline that corresponds to XMLTransmit.
11. If you chose **Static Solicit-Response Port** in step 4, specify send and receive pipelines.
 - a. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - b. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
12. Click **OK**.

To deploy adapters for receiving messages from Oracle Database

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleDB adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
4. Expand the application under which you want to deploy the Oracle Database adapter.
5. Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and the Oracle database.
6. In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
7. On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
8. In the **Receive Location Properties** dialog box, do the following:
 - a. Specify a name for the receive location.
 - b. From the **Type** drop-down list, select WCF-OracleDB, and then click **Configure**.
9. In the transport properties dialog box, do the following:
 - a. Click the **General** tab, click the **Configure** button, and provide values for the connection parameters. For more information about the connection URI, see [The Oracle Database Connection URI](#).
 - b. Click the **Binding** tab and specify values for binding properties exposed by the Oracle Database adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

 **Note**

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- c. Click the **Behavior** tab to set the transaction isolation level. For more information about setting transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).
- d. Click the **Others** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to the Oracle database.

- To connect using the Oracle database credentials, type the database credentials in the **User name** and **Password** text boxes.
 - To connect using Windows Authentication, type / in the **User name** text box and leave the **Password** text box blank.
- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.
- e. To return to the **Receive Location Properties** dialog box, click **OK**.
10. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.
 11. If you chose **One-way Receive Port** in step 5, specify a receive pipeline. From the **Receive pipeline** list, select the pipeline corresponding to XMLReceive.
 12. If you chose **Request Response Receive Port** in step 5, specify send and receive pipelines.
 - a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 13. In the **Receive Location Properties** dialog box, click **OK**.
 14. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Tasks

[Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#)

Concepts

[Connecting to the Oracle Database Using Windows Authentication](#)

Configuring a Physical Port Binding by Using a Port Binding File

When you use the Consume Adapter Service BizTalk Project Add-in to generate metadata for an Oracle database artifact, other than the schema files, the Consume Adapter Service Add-in also generates a port binding file. You can import this binding file into your BizTalk application to create a physical send or receive port. For instructions on importing binding files, see [Importing Bindings](#). If you import this binding file, you do not have to manually create a physical send or receive port.

Important

While using the Consume Adapter Service BizTalk Project Add-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file. You must manually add the binding property and its value in the binding file, if required.

Creating a port using a port binding file always creates a two-way send or receive ports. If you want to create one-way send or receive ports, you can create it manually by following the procedure mentioned in [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Or, you can follow the workarounds documented in this topic to modify the port binding file to create one-way send or receive ports.

Following are some key points that you must understand with respect to the binding file generated by the Consume Adapter Service Add-in:

- The files are created with a specific naming convention. If you generated metadata for outbound operations, that is to send messages to the Oracle database, the name of the file is `WcfSendPort_OracleDBBinding_Custom.bindinginfo.xml`.

If you generated metadata for inbound operations, that is to receive messages from the Oracle database, the name of the file is `WcfReceivePort_OracleDBBinding_Custom.bindinginfo.xml`.

- The file contains information about the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. When you import this binding file to create a port, all the relevant information required to configure a physical port is automatically set on the port.

Important

By default, the action on the send port is mapped to the operation name for which you generate metadata. For example, if you generate metadata for a Select operation on the ACCOUNTACTIVITY table, the action on the port is set to `<Operation Name="Select" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY/Select" />`. However, the operation name on the logical send port you create in the BizTalk orchestration might not be the same. You must ensure that the operation name in the logical port (in the BizTalk orchestration) and the physical send port (in BizTalk Server Administration console) are the same. If not, you will receive an error while sending messages to the Oracle database through the send port.

- You only need to provide the credentials for the port to connect to the Oracle database. While the binding file does retain the username with which to connect, for security reasons the binding file does not contain the password.

Key Considerations for Using the Port Binding File

- When you import the binding file, you might get a dialog message informing that the BizTalk application name in the binding file does not match the application name to which you are importing the binding file. You can safely ignore this message and continue.
- The binding file also contains names of ports and receive locations. If the BizTalk application to which you are importing the binding file creates a port or a receive location that has the same name as an already existing port in the same BizTalk application, you will get an error. You must manually edit the bindings file to specify a unique name for the ports or receive locations.
- The binding file also contains information about the connection URI. If the binding file creates a receive location that has

the same receive URI as an already existing receive location in the same BizTalk application, you will get an error. You must manually edit the binding file to specify a unique URI. You can specify a unique URI by including a polling ID.

- By default, the port binding file always contains definitions for two-way ports (send or receive). When you import this file in a BizTalk application, it creates a two-way send or receive port. However, you may have an orchestration that has a one-way send or receive port. So, when you configure such an orchestration and use the port created by importing the binding file, the port is not available in the list. This happens because the logical port you created as part of the orchestration is a one-way port while the physical port created in the orchestration is a two-way port. In such cases, you can edit the bindings file to make the following changes:

For this	Do this
<p>To edit the port bindings file to configure a one-way send port</p>	<ol style="list-style-type: none"> In the following excerpt, change the value of IsTwoWay property to false. Originally, this is set to true. <div data-bbox="703 600 1497 719" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><SendPort Name="port_name" IsStatic="true" IsTwoWay="false" BindingOption="0"></pre> </div> Comment out the following excerpts: <div data-bbox="703 808 1497 1144" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipeline Name="Microsoft.BizTalk.DefaultPipelines.XMLReceive" FullyQualifiedName="Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=token" Type="1" TrackingOption="None" Description=""/></pre> </div> <div data-bbox="703 1167 1497 1245" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipelineData xsi:nil="true" /></pre> </div>

To edit the port bindings file to configure a one-way receive port

1. In the following excerpt, change the value of **IsTwoWay** property to false. Originally, this is set to true.

```
<ReceivePort Name="port_name" IsTwoWay="false" BindingOption="1">
```

2. Comment out the following excerpts:

```
<SendPipeline Name="Microsoft.BizTalk.DefaultPipelines.XMLTransmit"
  FullyQualifiedName="Microsoft.BizTalk.DefaultPipelines.XMLTransmit,
  Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral,
  PublicKeyToken=token" Type="2" TrackingOption="None" Description="" />
```

```
<SendPipelineData xsi:nil="true" />
```

```
<SendPipelineData xsi:nil="true" />
```

Configuring a WCF-OracleDB Port Using the Port Binding File Generated Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in creates a port binding file that you can import in BizTalk Server Administration console. You can use the same port binding file to also create the BizTalk WCF-OracleDB port in BizTalk Server Administration console. However, before creating WCF-OracleDB port you must perform the following tasks to modify the port binding file.

1. Open the port binding file in a text editor.
2. Search and replace "WCF-Custom" with the name with which you added the WCF-OracleDB adapter in BizTalk Server Administration console. For example, if you added the WCF-OracleDB adapter as "OracleDBAdapter", replace "WCF-Custom" with "OracleDBAdapter".
3. Search for the "ConfigurationClsid" attribute, and replace the existing value of the attribute with "D7127586-E851-412e-8A8A-2428AEDDC219".
4. Save and close the binding file.
5. Import the binding file in BizTalk Server Administration console. For instructions on how to import the binding file, see [Importing Bindings](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring Dynamic Ports

In BizTalk Server, you can configure dynamic ports for a WCF-Custom adapter. Because the Oracle Database adapter is a WCF-based adapter, you can dynamically configure a port for the Oracle Database adapter by using message context properties.

For the Oracle Database adapter, the URI, action, and binding may be determined from a property on an incoming message, and then specified in the **Expression** shape, as shown in the following example:

```
Request2=Request1;
Request2(WCF.Action)="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTI
VITY/Select";
Request2(WCF.BindingType)="oracleDBBinding";
Request2(WCF.UserName)="SCOTT";
Request2(WCF.Password)="TIGER";
SendPort(Microsoft.XLANGs.BaseTypes.Address)="oracledb://adapdoc/";
SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="WCF-Custom";
```

Note

If you are using a WCF-OracleDB adapter in BizTalk Server Administration console, you can also specify the transport type as `SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="OracleDatabaseAdapter"`, where **OracleDatabaseAdapter** is the name with which you added the WCF-OracleDB adapter in BizTalk Server Administration console.

In the above example,

- Request2 message is being created from Request1 message. Both the messages map to an operation schema, which is generated using the Consume Adapter Service BizTalk Project Add-in.
- SendPort is the name of the logical send port in the BizTalk orchestration.

The **Expression** shape is part of the BizTalk orchestration. When you deploy the orchestration, the WCF-custom send port is also created.

For more information on configuring dynamic ports, see <http://go.microsoft.com/fwlink/?LinkId=106735>.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Reusing Adapter Bindings

A binding creates a mapping between a logical endpoint (such as an orchestration port or a role link) and a physical endpoint (such as a send and receive port). This enables communication between different components of a BizTalk business solution. You can create bindings by using the BizTalk Server Administration console.

What Is a Binding File?

A binding file is an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. The binding file describes:

- The host to which each orchestration is bound.
- The trust level of the host.
- The settings for each send port, receive port, receive location, and party that has been configured.

You can generate binding files and then apply the bindings that they contain to an assembly, application, or group. This prevents having to manually configure bindings in different deployment environments and speeds up application deployment.

A binding file is not automatically generated for a BizTalk assembly, application, or group. However, you can generate a binding file by exporting bindings, as described in [Exporting Bindings](#). You can then import the binding file into an application or group, as described in [Importing Bindings](#).

For more information about bindings and about binding files, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

In This Section

- [Exporting Bindings](#)
- [Importing Bindings](#)

Exporting Bindings

This section describes how to export bindings for a BizTalk application into an XML file. You can then import the bindings from the XML file into another BizTalk application. Importing bindings overwrites any existing bindings of the same name in the application. You can also add bindings to an application, which does not overwrite existing bindings. The bindings that you add do not take effect until you import the application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To export bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application whose bindings you want to export, point to **Export**, and then click **Bindings**.
4. On the **Export Bindings** page, in **Export to file**, type the absolute path of the XML file to which to export the bindings.

Example: **C:\Bindings\Application1Bindings.Binding1.xml**

5. Ensure that **Export all bindings from the current application** is selected.
6. To export all party information for the group, select the **Export Global Party information** check box.
7. Click **OK**.

The bindings are exported into an XML file in the location that you specified.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the Transport Properties dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for the Oracle Database](#).

See Also

Tasks

[Importing Bindings](#)

Concepts

[Reusing Adapter Bindings](#)

Importing Bindings

This topic describes how to use the BizTalk Server Administration console to import bindings into a BizTalk application from an XML file.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To import bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application into which you want to import bindings, point to **Import**, and then click **Bindings**.
4. Click the binding file, and then click **Open**.

The artifacts in the binding file are written to the application. They display in appropriate folders of the application. For example, the send ports imported as part of the bindings display under the **Send Ports** folder.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the Transport Properties dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for the Oracle Database](#).

See Also

Tasks

[Exporting Bindings](#)

Concepts

[Reusing Adapter Bindings](#)

Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server

The Microsoft BizTalk Adapter for Oracle Database surfaces a set of standard operations on Oracle database tables and views. These are called data manipulation language operations (DML) operations using which you can perform simple INSERT, UPDATE, SELECT, and DELETE statements qualified by a WHERE clause on tables and views. For more information about how the adapter supports these operations, see [Performing Basic Insert, Update, Delete, and Select Operations on Oracle Tables and Views](#). For information about the structure of the SOAP message for DML operations, see [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#).

Note

To perform more complex operations, for example running a parameterized SQL SELECT query, you can use the SQLEXECUTE operation. For more information about using the SQLEXECUTE operation with BizTalk Server, see [Performing a SQLEXECUTE Operation by Using BizTalk Server](#).

How to Perform Basic Operations on an Oracle Database?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform Insert, Update, Delete, or Select operations on tables and views in an Oracle database, these tasks are:

1. Create a BizTalk project and generate schema for the operation you want to invoke on an Oracle database table or view.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the operation on the Oracle database table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, SelectAccTable, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to perform basic DML operations, we will select records from the ACCOUNTACTIVITY table under the SCOTT schema in the Oracle database. This table is created under the SCOTT schema by running the SQL scripts provided with the samples. To know more about the samples, see [Samples](#).

To demonstrate how to select records, we generate schema for Select operation for the ACCOUNTACTIVITY table under the SCOTT schema. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Oracle database and the other to receive a response.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and then select <i>SelectAccTable.OracleDBBindingSchema.Select</i> , where <i>SelectAccTable</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the Select operation on the ACCOUNTACTIVITY table.

5. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

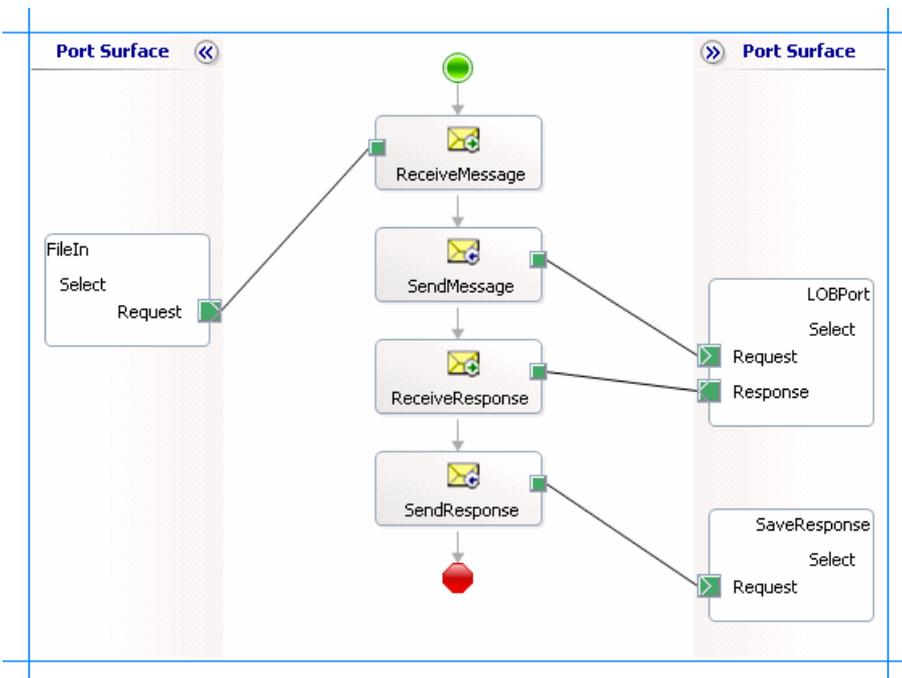
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and then select <i>SelectAccTable.OracleDBBindingSchema.SelectResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on the Oracle database. In this orchestration, you drop a request message at a defined receive location. The Oracle Database adapter consumes this message and passes it on to the Oracle database via ODP. The response from the Oracle database is saved to another location. A typical orchestration for performing basic table operations on Oracle database would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration for the Select operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> Set Identifier to <i>FileIn</i> Set Type to <i>FileInPort</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Select.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Select.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Select.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Select.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console.

For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
- Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for selecting records from an Oracle database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the Select operation you generated earlier. For example, the request message to select records from ACCOUNTACTIVITY table having the ACCOUNT field equal to 100001 is:

```
<Select xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY">  
  <COLUMN_NAMES>*</COLUMN_NAMES>  
  <FILTER>ACCOUNT=100001</FILTER>  
</Select>
```

See [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#) for more information about the request message schema for performing basic DML operations on Oracle database tables and views using the Oracle Database adapter.

The orchestration consumes the message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the above request message is:

```
<?xml version="1.0" encoding="utf-8" ?>  
  <SelectResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY">  
    <SelectResult>  
      <ACCOUNTACTIVITYRECORDSELECT>  
        <TID>1</TID>  
        <ACCOUNT>100001</ACCOUNT>  
        <AMOUNT>500</AMOUNT>
```

```
<DESCRIPTION />
<TRANSDATE>2007-10-16T16:58:44</TRANSDATE>
<PROCESSED>n</PROCESSED>
</ACCOUNTACTIVITYRECORDSELECT>
<ACCOUNTACTIVITYRECORDSELECT>
  ...
  ...
<ACCOUNTACTIVITYRECORDSELECT>
  ...
  ...
</SelectResult>
</SelectResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while performing a DML operation using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Tables with Large Object Types Data by Using BizTalk Server

The Microsoft BizTalk Adapter for Oracle Database provides support for the Oracle large object (LOB) data types:

- Binary large object (BLOB)
- Character large object (CLOB)
- National character large object (NCLOB)
- Binary file (BFILE). For more information, see [Performing Operations on Tables with BFILE Data Types by Using BizTalk Server](#).

The Microsoft BizTalk Adapter for Oracle Database does this by surfacing the ReadLOB and UpdateLOB operations for tables that contain LOB columns. For more information about these operations see [Operations on Tables and Views That Contain LOB Data](#). For more information about the structure of the SOAP message for invoking these operations, see [Message Schemas for Special LOB Operations](#).

Note

When using the Oracle Database adapter with BizTalk Server, the ReadLOB operation does not support streaming LOB type data from an Oracle database. To stream LOB data from an Oracle database using BizTalk Server you should instead use the Select operation. For more information about streaming, see [Streaming Support for LOB Data Types](#). Also, the response from the Oracle database for the ReadLOB operation will fail validation against the WSDL. For instructions on how to work around the failure, see [Troubleshooting Operational Issues](#).

How to Perform Operations on LOB Data?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke both ReadLOB and UpdateLOB operations on a table in an Oracle database, these tasks are:

1. Create a BizTalk project and generate schema for both ReadLOB and UpdateLOB operations.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database. You must create messages for both the sending request and receiving responses for both the operations.
3. Create an orchestration to invoke both ReadLOB and UpdateLOB operations.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, Operate_LOB, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to perform ReadLOB and UpdateLOB operations, we will generate metadata for these operations surfaced for the CUSTOMER table under the SCOTT schema in the Oracle database. This table is created under the SCOTT schema by running the SQL scripts provided with the samples. To know more about the samples, see [Samples](#).

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two request-response message sets—one request-response set for the ReadLOB operation and the second request-response set for the UpdateLOB operation.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do this, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message, and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>Operate_LOB.OracleDBBindingSchema.ReadLOB</i> , where <i>Operate_LOB</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the ReadLOB and UpdateLOB operations on the CUSTOMER table.

5. Repeat the previous step to create three more messages. In the **Properties** pane for the new messages, do the following:

Set Identifier to	Set Message Type to
Response	<i>Operate_LOB.OracleDBBindingSchema.ReadLOBResponse</i>
Request2	<i>Operate_LOB.OracleDBBindingSchema.UpdateLOB</i>
Response2	<i>Operate_LOB.OracleDBBindingSchema.UpdateLOBResponse</i>

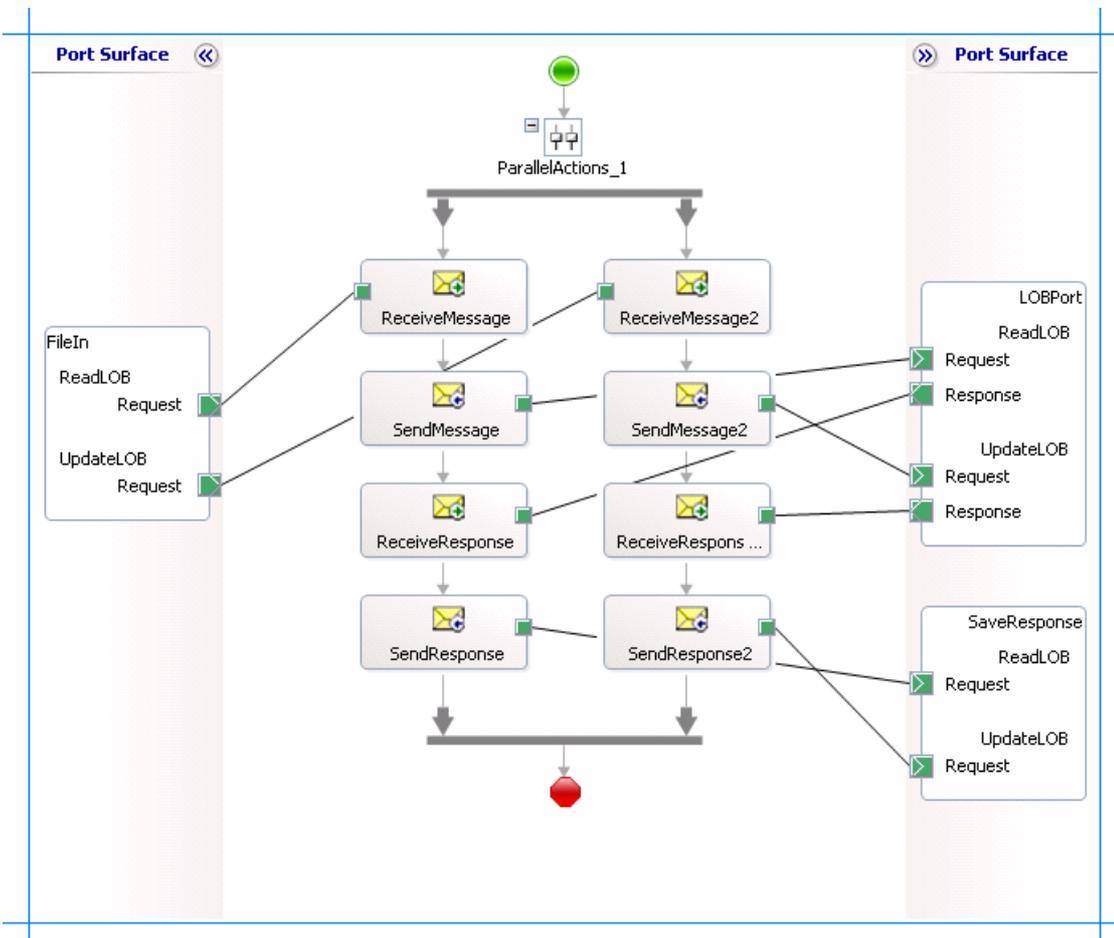
Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking ReadLOB and UpdateLOB operations on a table. In this orchestration, you drop two request messages, one for the ReadLOB operation and the other for the UpdateLOB operation. These messages are dropped at a receive location. The Oracle Database adapter consumes the messages and passes them on to the Oracle database via ODP. The response from the Oracle database is saved to another location.

Because the orchestration picks two requests simultaneously, you need to include a Parallel Actions shape in the orchestration. For each parallel action, you must include Send and Receive shapes to send messages to Oracle database and receive responses. However, you could use the same ports for sending and receiving messages for both operations. A typical orchestration for performing the ReadLOB and UpdateLOB operations simultaneously would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration. The following table lists the shapes you must include for one of the parallel actions.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

The following table lists the shapes that you must include for the other parallel action.

Shape	Shape Type	Properties
ReceiveMessage2	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage2</i> Set Activate to <i>True</i>
SendMessage2	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage2</i>

ReceiveResponse2	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse2</i> • Set Activate to <i>False</i>
SendResponse2	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse2</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Because you will be processing two request and response messages using these ports, you must create two operations for each port, where each operation corresponds to one message type. To create an operation, right-click the port shape, and then select **New Operation**. Name the first operation for each port as **ReadLOB** and the second operation for each port as **UpdateLOB**.

Using Correlation

Correlation is the process of matching an incoming message with the appropriate instance of an orchestration. In the orchestration you will be dropping two request messages, one for each overload. Using correlation, you associate a request message with the right orchestration. For more information about correlation, see <http://go.microsoft.com/fwlink/?LinkId=103055>.

To use correlations

1. Promote a property from the schema generated for each operation. For example, promote the LOB_COLUMN property from the ReadLOB operation schema; promote the FILTER property from the UpdateLOB operation schema. To promote a property, right-click the property in the schema view, point to **Promote**, and then select **Quick Promotion**. This adds a PropertySchema.xsd file to your BizTalk project.

For information about promoting a property, see <http://go.microsoft.com/fwlink/?LinkId=102358>.

2. From the Orchestration View, right-click **Correlation Types**, and then select **New Correlation Type**.
3. The **Correlation Properties** dialog box lists the properties you promoted in step 1. Select a property, and then click **Add**.
4. Click **OK**.
5. To create correlation types for the other promoted property, repeat these steps.
6. Rename the correlation types based on the operation they are associated with. You could rename the correlation types to *CorrelationType_ReadLOB* (for the ReadLOB operation) and *CorrelationType_UpdateLOB* (for the UpdateLOB operation).
7. From Orchestration View, right-click **Correlation Sets**, and then select **New Correlation Set**.
8. Right-click the newly added correlation set, and then click **Properties**. In the **Properties** pane, do the following:

Use this	To do this
Correlation Type	<i>Operate_LOB.CorrelationType_ReadLOB</i>
Identifier	<i>Correlation_ReadLOB</i>

9. Add another correlation set and specify the following properties from the Properties pane.

Use this	To do this
Correlation Type	<i>Operate_LOB.CorrelationType_UpdateLOB</i>
Identifier	<i>Correlation_UpdateLOB</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link them to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_ReadLOB</i> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.ReadLOB.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.ReadLOB.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.ReadLOB.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.ReadLOB.Request</i>

ReceiveMessage2	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_UpdateLOB</i> • Set Message to <i>Request2</i> • Set Operation to <i>FileIn.UpdateLOB.Request</i>
SendMessage2	<ul style="list-style-type: none"> • Set Message to <i>Request2</i> • Set Operation to <i>LOBPort.UpdateLOB.Request</i>
ReceiveResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response2</i> • Set Operation to <i>LOBPort.UpdateLOB.Response</i>
SendResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.UpdateLOB.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop the request messages, one each for ReadLOB and UpdateLOB operations. The BizTalk orchestration will consume the request messages and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response messages, one for each operation, containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Because the WCF-Custom or WCF-OracleDB send port sends and receive messages conforming to more than one schema and performs two operations, you must set dynamic action for both the operations. For more information about actions, see [Specifying SOAP Action](#). For this orchestration, the action should be set as follows:

```
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="ReadLOB" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/ReadLOB" />
  <Operation Name="UpdateLOB" Action="http://Microsoft.LobServices.OracleDB/2007/
```

```
03/SCOTT/Table/CUSTOMER/UpdateLOB" />
</BtsActionMapping>
```

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for performing an operation on the Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom send port or WCF-OracleDB to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop request messages to the FILE receive location. The schema for the request messages must conform to the schema for the operations you generated earlier. See [Message Schemas for Special LOB Operations](#) for more information about the request message schema for invoking operations on LOB data types using the Oracle Database adapter.

The orchestration consumes the request messages and sends them to the Oracle database. The response from the Oracle database is saved at other FILE location defined as part of the orchestration.

For this orchestration, we first drop a request message for the UpdateLOB operation to update the PHOTO column (of BLOB data type) of the CUSTOMER table. The request message to invoke the update the PHOTO column for a specific customer is:

```
<UpdateLOB xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER">
  <LOB_COLUMN>PHOTO</LOB_COLUMN>
  <FILTER>Name='Mindy Martin'</FILTER>
  <Stream>YWJjZA==</Stream>
</UpdateLOB>
```

Note

The filter string must always fetch one matching row otherwise the Oracle Database adapter throws an XmlReaderParsingException. Also the value for the <Stream> element must be of base64Binary type.

The response for the UpdateLOB operation is:

```
<?xml version="1.0" encoding="utf-8"?>
<UpdateLOBResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER"></UpdateLOBResponse>
```

We now drop a request message for the ReadLOB operation to read the data that was updated by the UpdateLOB operation. The request message to invoke the ReadLOB operation on the PHOTO column for a specific customer is:

```
<ReadLOB xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER">
  <LOB_COLUMN>PHOTO</LOB_COLUMN>
```

```
<FILTER>NAME='Mindy Martin'</FILTER>
</ReadLOB>
```

Note

The filter string must always fetch one matching row. If there is more than one matching row, the Oracle Database adapter only returns the LOB column for the first (matching) row.

The response for the ReadLOB operation is:

```
<?xml version="1.0" encoding="utf-8"?>
<ReadLOBResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER">
  <ReadLOBResult>YWJjZA==</ReadLOBResult>
</ReadLOBResponse>
```

Note

The response for the ReadLOB operation may fail to validate against the WSDL. You must perform certain tasks to validate the ReadLOB against the WSDL. For more information see [Troubleshooting Operational Issues](#).

Possible Exceptions

For information about the exceptions you might encounter while performing operations on table containing LOB data using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Functions and Procedures by Using BizTalk Server

Adapter clients can invoke functions and procedures in an Oracle database by using the Oracle Database adapter with BizTalk Server. The Oracle Database adapter surfaces procedures, functions, and packages as operations. The Oracle Database adapter also enables adapter clients to invoke:

- Overloaded functions and stored procedures. See [Invoking Overloaded Functions and Procedures by Using BizTalk Server](#).
- Functions and procedures having IN, OUT, and IN OUT REF CURSOR parameters. See [Invoking Functions and Procedures with REF CURSORS by Using BizTalk Server](#).
- Functions and procedures having IN, OUT, and IN OUT RECORD type parameters. See [Invoking Functions and Procedures with RECORD Types by Using BizTalk Server](#).

For more information about working with functions and stored procedures by using the Oracle Database adapter, see [Operations on Functions and Stored Procedures](#). For information about the structure of the SOAP message for invoking functions and procedures, see [Message Schemas for Functions and Procedures](#).

How to Invoke Functions in an Oracle Database?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a function in an Oracle database, these tasks are:

1. Create a BizTalk project and generate schema for the function you want to invoke in an Oracle database.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the function in the Oracle database.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `InvokeFunction`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to invoke a function, we will invoke the `CREATE_ACCOUNT` function under the `SCOTT\Package\ACCOUNT_PKG` schema. This package is created under the `SCOTT` schema by running the SQL scripts provided with the samples. The `CREATE_ACCOUNT` function takes account and addresses information as inputs and creates records in the `ACCOUNT` and `CUSTOMER` tables. If the record already exists, the function returns zero; otherwise, the function returns the account ID. To know more about the samples and the SQL scripts, see [Samples](#).

To invoke the `CREATE_ACCOUNT` function, we must generate schema for the same function under the `SCOTT\Package\ACCOUNT_PKG` schema. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Oracle database and the other to receive a response.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message, and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and then select <i>InvokeFunction.OracleDBBindingSchema.CREATE_ACCOUNT</i> , where <i>InvokeFunction</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the CREATE_ACCOUNT function.

5. Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

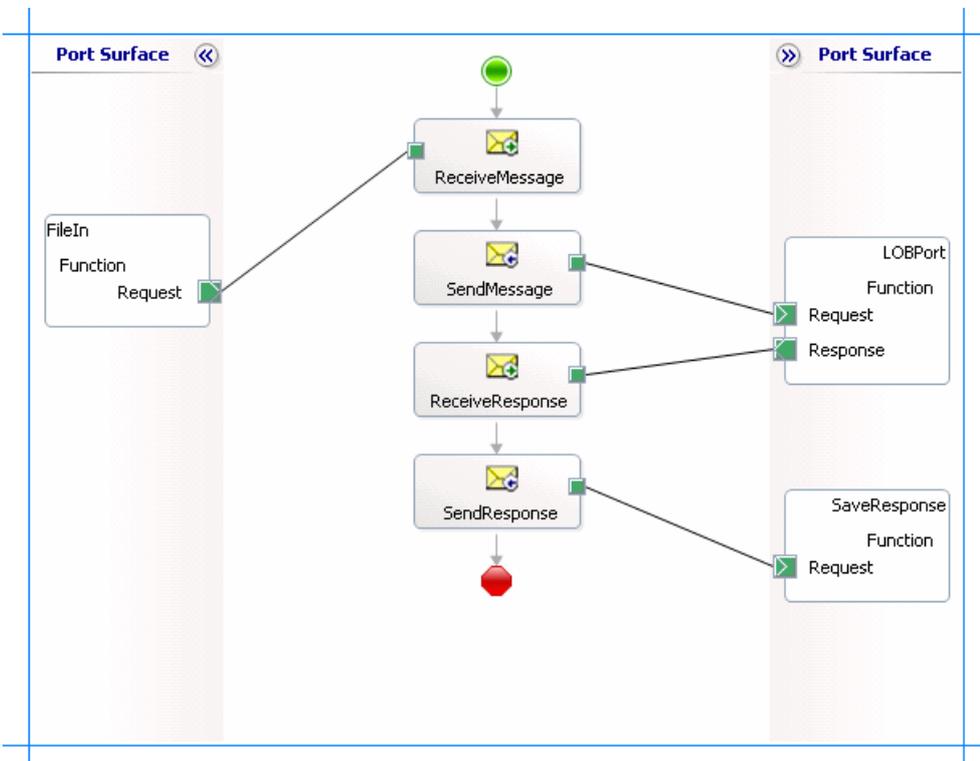
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and then select <i>InvokeFunction.OracleDBBindingSchema.CREATE_ACCOUNTResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking functions on an Oracle database. In this orchestration, you drop a request message at a defined receive location. The Oracle Database adapter consumes this message and passes it on to the Oracle database via ODP. The response from the Oracle database is saved to another location. A typical orchestration for invoking functions and procedures in an Oracle database would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> Set Identifier to <i>FileIn</i> Set Type to <i>FileInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Function.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Function.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Function.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Function.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console.

For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
- Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking a function in an Oracle database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the function you generated earlier. See [Message Schemas for Functions and Procedures](#) for more information about the request message schema for invoking functions using the Oracle Database adapter.

For example, the request message to invoke the CREATE_ACCOUNT function is:

```
<CREATE_ACCOUNT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <ACCT>
    <ACCTID xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT"></ACCTID>
    <NAME xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">John Smith</NAME>
    <BALANCE xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">10000</BALANCE>
  </ACCT>
  <ADDR>
    <STREET xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">BelRed Road</STREET>
    <CITY xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">Redmond</CITY>
    <STATE xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">WA</STATE>
  </ADDR>
</CREATE_ACCOUNT>
```

The orchestration consumes the message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the above request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<CREATE_ACCOUNTResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
<CREATE_ACCOUNTResult>[ID]</CREATE_ACCOUNTResult>
</CREATE_ACCOUNTResponse>
```

Where, [ID] is the ID of the account created by the function.

Possible Exceptions

For information about the exceptions you might encounter while invoking functions and procedures using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Overloaded Functions and Procedures by Using BizTalk Server

Stored procedures and functions can be overloaded in an Oracle database. The Oracle Database adapter supports overloaded functions and procedures by changing the target namespace of the operation. For example, the message structure for two overloaded procedures looks like:

```

Stored Procedure Overload 1:
<[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/overload1">
  <[PRM1_NAME]>value1</[PRM1_NAME]>
  <[PRM2_NAME]>value1</[PRM2_NAME]>
  ...
</[SP_NAME]>

Stored Procedure Overload 2:
<[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/overload2">
  <[PRM1_NAME]>value1</I_[PRM1_NAME]>
  <[PRM2_NAME]>value1</I_[PRM2_NAME]>
  ...
</[SP_NAME]>

```

The SOAP message structure and the SOAP action required to invoke an overloaded function or procedure is similar to invoking a function and procedure, as described under [Message Schemas for Functions and Procedures](#).

Invoking an overloaded procedure is similar to invoking any other function as described in [Invoking Functions and Procedures by Using BizTalk Server](#). However to differentiate overloaded functions, the Oracle Database adapter appends a unique string to the node ID and the namespace that it surfaces for overloaded artifact. This string is "overload1" for the first overload, "overload2" for the next overload, and so on.

How to Invoke Overloaded Functions and Procedures?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a function in an Oracle database, these tasks are:

1. Create a BizTalk project and generate schema for the overloaded function you want to invoke in an Oracle database.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database. You must create messages for each overload.
3. Create an orchestration to invoke the overloaded function in the Oracle database.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `InvokeOverloadedProc`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to invoke an overloaded function or procedure, we will invoke the `GET_ACCOUNT` procedure under the `SCOTT\Package\ACCOUNT_PKG` schema. This package is created under the `SCOTT` schema by running the SQL scripts provided with the samples. This is an overloaded procedure where:

- One overload takes the account ID as the IN parameter and returns an ACCOUNT%ROWTYPE as OUT parameter.
- Second overload takes the account name as the IN parameter and returns an ACCOUNT%ROWTYPE as OUT parameter.

To know more about the samples and the SQL scripts, see [Samples](#).

To invoke an overloaded function, we generate schema for both the overloaded procedures, GET_ACCOUNT.1 and GET_ACCOUNT.2. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two request-response message sets—one request-response set for the first overloaded procedure and the second request-response set for the second overloaded procedure.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message, and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>InvokeOverloadedProc.OracleDBBindingSchema.GET_ACCOUNT</i> , where <i>InvokeOverloadedProc</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the GET_ACCOUNT procedure.

5. Repeat the previous step to create three more messages. In the **Properties** pane for the new messages, do the following:

Set Identifier to	Set Message Type to
Response	<i>InvokeOverloadedProc.OracleDBBindingSchema.GET_ACCOUNTResponse</i>
Request2	<i>InvokeOverloadedProc.OracleDBBindingSchema1.GET_ACCOUNT</i>
Response2	<i>InvokeOverloadedProc.OracleDBBindingSchema1.GET_ACCOUNTResponse</i>

Setting up the Orchestration

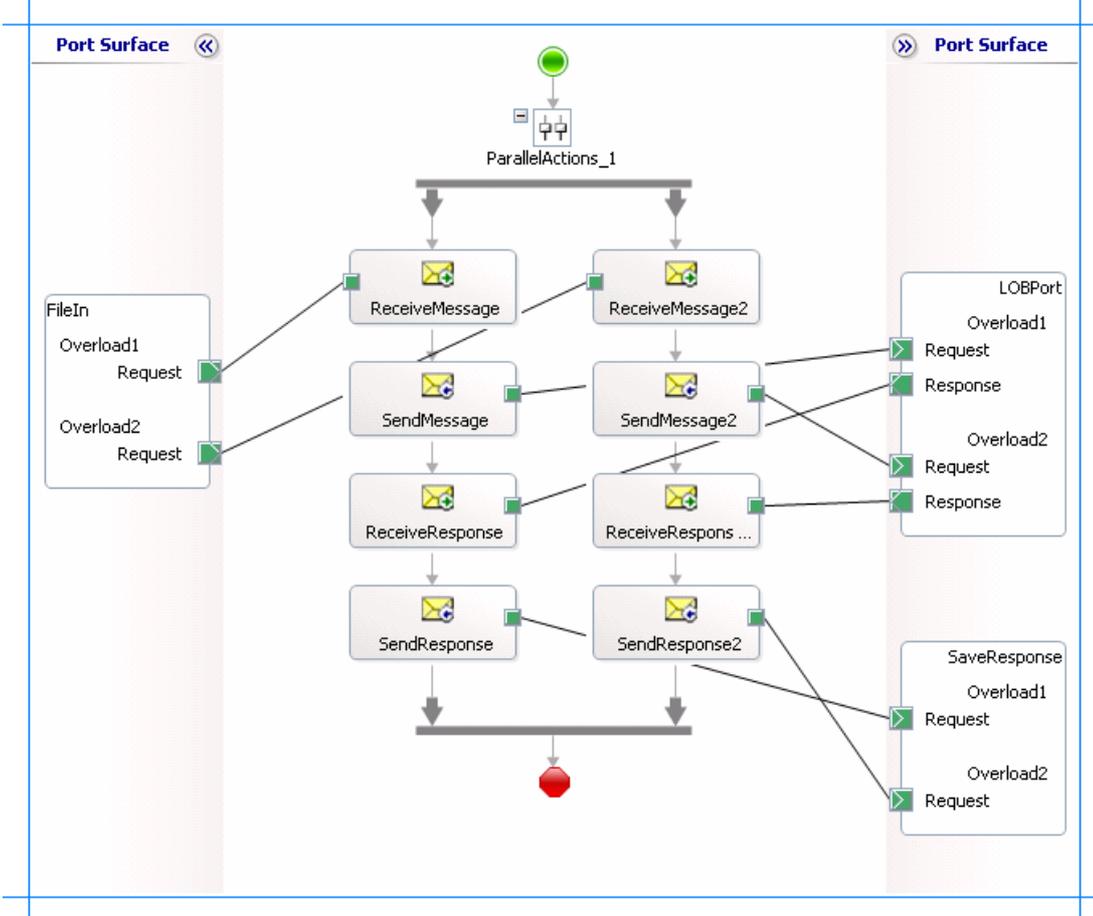
You must create a BizTalk orchestration to use BizTalk Server for invoking an overloaded procedure in an Oracle database. In this orchestration, you drop two request messages, one corresponding to each overloaded procedure, at the defined receive location. The Oracle Database adapter consumes the messages and passes them on to the Oracle database via ODP. The response from the Oracle database is saved to another location.

Because the orchestration picks two requests simultaneously, you need to include a Parallel Actions shape in the orchestration. For each parallel action, you must include Send and Receive shapes to send messages to the Oracle database and receive responses. However, you could use the same ports for sending and receiving messages for both operations. A typical orchestration for invoking the overloaded procedures simultaneously would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.

- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration to invoke the first and second overloads for GET_ACCOUNT procedure resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration. The following table lists the shapes you must include for one of the parallel actions.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

The following table lists the shapes that you must include for the other parallel action.

Shape	Shape Type	Properties
ReceiveMessage2	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage2</i> • Set Activate to <i>True</i>
SendMessage2	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage2</i>
ReceiveResponse2	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse2</i> • Set Activate to <i>False</i>
SendResponse2	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse2</i>

Adding Ports

Make sure that you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Because you will be processing two request and response messages using these ports, you must create two operations for each port, where each operation corresponds to one message type. To create an operation, right-click the port shape, and then select **New Operation**. Name the first operation for each port as **Overload1** and the second operation for each port as **Overload2**.

Using Correlation

Correlation is the process of matching an incoming message with the appropriate instance of an orchestration. In the orchestration you will be dropping two request messages, one for each overload. Using correlation, you associate a request message with the right orchestration. For more information about correlation, see

<http://go.microsoft.com/fwlink/?LinkId=103055>.

To use correlations

1. Promote a property from the schema generated for each overloaded function. For example, promote the AID property from the schema for the first overload; promote the ANAME property from the schema of the second overload. To promote a property, right-click the property in the schema view, point to **Promote**, and then select **Quick Promotion**. This adds a PropertySchema.xsd file to your BizTalk project.

For information about promoting a property, see <http://go.microsoft.com/fwlink/?LinkId=102358>.

2. From Orchestration View, right-click **Correlation Types**, and then select **New Correlation Type**.
3. The **Correlation Properties** dialog box lists the properties that you promoted in step 1. Select a property, and then click **Add**.
4. Click **OK**.
5. To create correlation types for the other promoted property, repeat these steps.
6. Rename the correlation types based on the property to which they are associated. You could rename the correlation types to *CorrelationType_AID* (for the AID property) and *CorrelationType_ANAME* (for the ANAME property).
7. From Orchestration View, right-click **Correlation Sets**, and then select **New Correlation Set**.
8. Right-click the newly added correlation set, and then click **Properties**. In the Properties pane, do the following:

Use this	To do this
Correlation Type	<i>InvokeOverloadedProc.CorrelationType_AID</i>
Identifier	<i>Correlation_AID</i>

9. Add another correlation set, and specify the following properties from the Properties pane.

Use this	To do this
Correlation Type	<i>InvokeOverloadedProc.CorrelationType_ANAME</i>
Identifier	<i>Correlation_ANAME</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_AID</i> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Overload1.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Overload1.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Overload1.Response</i>

SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Overload1.Request</i>
ReceiveMessage2	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_ANAME</i> • Set Message to <i>Request2</i> • Set Operation to <i>FileIn.Overload2.Request</i>
SendMessage2	<ul style="list-style-type: none"> • Set Message to <i>Request2</i> • Set Operation to <i>LOBPort.Overload2.Request</i>
ReceiveResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response2</i> • Set Operation to <i>LOBPort.Overload2.Response</i>
SendResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response2</i> • Set Operation to <i>SaveResponse.Overload2.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop the request messages, one for each overloaded procedure. The BizTalk orchestration will consume the request messages and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response messages, one for each overloaded procedure, containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Because the WCF-Custom or WCF-OracleDB send port sends and receive messages conforming to more than one schema and performs two operations, you must set dynamic action for both the operations. For more information about actions, see [Specifying SOAP Action](#). For this orchestration, the action should be set as follows:

```
<BtsActionMapping>
```

```
<Operation Name="Overload1" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT/overload1" />
<Operation Name="Overload2" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT/overload2" />
</BtsActionMapping>
```

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking a function in an Oracle database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop two request messages (one for each overloaded procedure) to the FILE receive location. The schema for the request messages must conform to the schema for the procedure you generated earlier. See [Message Schemas for Functions and Procedures](#) for more information about the request message schema for invoking functions using the Oracle Database adapter.

For example, the request message to invoke the first overload of the GET_ACCOUNT procedure is:

```
<GET_ACCOUNT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT/overload1">
  <AID>100001</AID>
</GET_ACCOUNT>
```

Similarly, the request message to invoke the second overload of the GET_ACCOUNT procedure is:

```
<GET_ACCOUNT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT/overload2">
  <ANAME>Mindy Martin</ANAME>
</GET_ACCOUNT>
```

The first request message invokes the GET_ACCOUNT procedure to retrieve record with account ID equal to 100020. The second request message invokes the GET_ACCOUNT procedure to retrieve records having the account name as "John Smith".

The orchestration consumes the request messages and sends them to the Oracle database. The response from the Oracle database is saved at other FILE location defined as part of the orchestration. For example, the response for invoking the first overloaded procedure is:

```
<?xml version="1.0" encoding="utf-8"?>
<GET_ACCOUNTResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT/overload1">
  <ACCT>
    <ACCTID>100001</ACCTID>
```

```
<NAME>Ty Carlson</NAME>  
<BALANCE>9000</BALANCE>  
</ACCT>  
</GET_ACCOUNTResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while invoking overloaded packages using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Functions and Procedures with REF CURSORS by Using BizTalk Server

A REF CURSOR is a PL/SQL data type that represents a pointer to a server-side result set generated by executing a query. A REF CURSOR type enables input and output streaming of data and is ideal for transferring large amounts of data to and from a PL/SQL code. The Oracle Database adapter provides support for strongly-typed and weakly-typed (SYS_REFCURSOR) REF CURSORS that can be passed to PL/SQL procedures and functions as IN, OUT, or IN OUT parameters. For more information about how the Oracle Database adapter supports REF CURSORS, see [Operations on Functions and Procedures with REF CURSOR Parameters](#). For information about the XML structure for REF CURSORS, see [Message Schemas for REF CURSORS](#).

How to Invoke Functions in an Oracle Database?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a function in an Oracle database that takes REF CURSOR as in parameters and gives REF CURSOR as out parameters, these tasks are:

1. Create a BizTalk project and generate schema for the function you want to invoke in an Oracle database.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the function in the Oracle database.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `Func_RefCursor`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how the Oracle Database adapter supports invoking function that takes REF CURSOR parameters we will invoke the `GET_ACTIVITY` procedure. This procedure takes a weakly-typed IN REF CURSOR and a strongly-typed IN OUT REF CURSOR as parameters. The function returns a status, a weakly-typed OUT REF CURSOR and a strongly-typed IN OUT REF CURSOR. The `GET_ACTIVITY` procedure is available as part of the `ACCOUNT_PKG` created by running the SQL scripts provided with the samples. To know more about the samples and the SQL scripts, see [Samples](#).

So, to invoke the `GET_ACTIVITY` procedure, we generate schema for the same procedure under the `SCOTT\Package\ACCOUNT_PKG` schema. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Oracle database and the other to receive a response.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do this, click **View**, point to **Other**

Windows, and then click **Orchestration View**.

2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message, and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>Func_RefCursor.OracleDBBindingSchema.GET_ACTIVITY</i> , where <i>Func_RefCursor</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the GET_ACTIVITY procedure.

5. Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

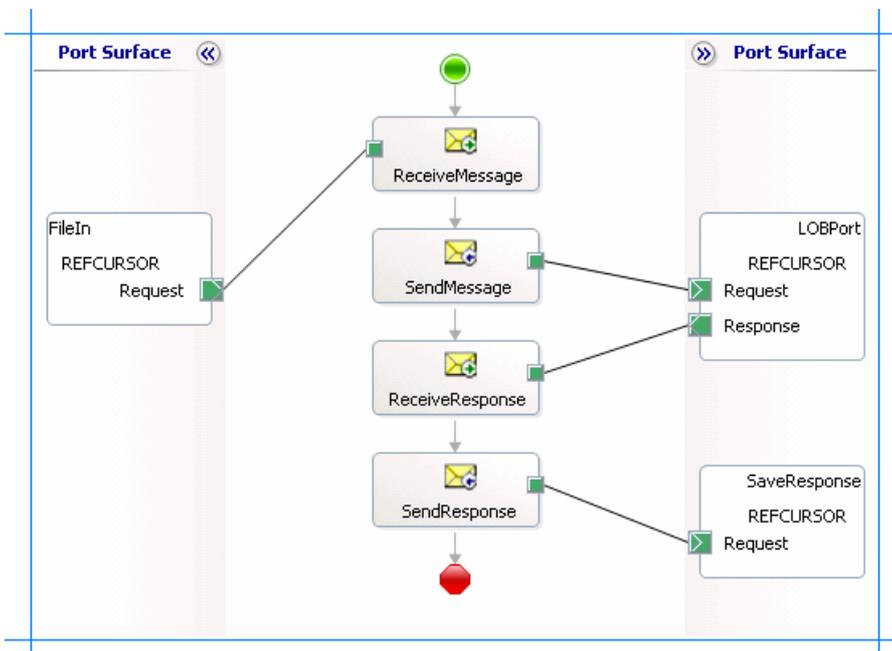
Use this	To do this
Identifier	Type Response .
Message Type	<i>Func_RefCursor.OracleDBBindingSchema.GET_ACTIVITYResponse</i>

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking a procedure with REF CURSOR parameters. In this orchestration, you drop a request message at a defined receive location. The Oracle Database adapter consumes this message and passes it on to the Oracle database via ODP. The response from the Oracle database is saved to another location. A typical orchestration for invoking functions and procedures with REF CURSORS would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> Set Identifier to <i>FileIn</i> Set Type to <i>FileInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> Set Identifier to <i>LOBPort</i> Set Type to <i>LOBPortType</i> Set Communication Pattern to <i>Request-Response</i> Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> Set Identifier to <i>SaveResponse</i> Set Type to <i>SaveResponseType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link them to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.REFCURSOR.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.REFCURSOR.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.REFCURSOR.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.REFCURSOR.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking a procedure in an Oracle database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the procedure you generated earlier. See [Message Schemas for Functions and Procedures](#) for more information about the request message schema for invoking functions using the Oracle Database adapter.

To invoke the GET_ACTIVITY procedure, you must specify a weakly-typed IN REF CURSOR and a strongly-typed IN OUT REF CURSOR as parameters. So, the request message to invoke this procedure is:

```
<GET_ACTIVITY xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <INRECS>BEGIN OPEN ? FOR SELECT * FROM ACCOUNTACTIVITY WHERE ACCOUNT=100001; END;</INRECS
>
  <INOUTRECS_IN>BEGIN ACCOUNT_PKG.GET_ALL_ACTIVITY(?); END;</INOUTRECS_IN>
</GET_ACTIVITY>
```

The orchestration consumes the request message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration.

The response for the above request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<GET_ACTIVITYResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <STATUS>5</STATUS>
  <INOUTRECS>
    <INOUTRECSRECORD xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACTIVITY">
      <TID>1</TID>
      <ACCOUNT>100001</ACCOUNT>
      <AMOUNT>500</AMOUNT>
      <DESCRIPTION />
      <TRANSDATE>2007-10-16T16:58:44</TRANSDATE>
      <PROCESSED>n</PROCESSED>
    </INOUTRECSRECORD>
    <INOUTRECSRECORD xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACTIVITY">
      . . . . .
      . . . . .
    </INOUTRECSRECORD>
    . . . . .
    . . . . .
  </INOUTRECS>
  <OUTRECS>
    <GenRecordRow xmlns="http://Microsoft.LobServices.OracleDB/2007/03">
      <GenRecordColumn>
        <GenRecordColumn>
          <ColumnName>TID</ColumnName>
          <ColumnValue>1</ColumnValue>
          <ColumnType>System.Decimal</ColumnType>
        </GenRecordColumn>
        <GenRecordColumn>
          . . . . .
        </GenRecordColumn>
        . . . . .
        . . . . .
      </GenRecordRow>
    </OUTRECS>
  </GET_ACTIVITYResponse>
```

```
    </GenRecordColumn>
  </GenRecordRow>
  <GenRecordRow xmlns="http://Microsoft.LobServices.OracleDB/2007/03">
    .....
    .....
  </GenRecordRow>
  .....
  .....
</OUTRECS>
</GET_ACTIVITYResponse>
```

Note that the response contains a status, a weakly-typed OUT REF CURSOR and a strongly-typed IN OUT REF CUROS.

Possible Exceptions

For information about the exceptions you might encounter while invoking functions and procedures using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Functions and Procedures with RECORD Types by Using BizTalk Server

Oracle RECORD types are used to represent hierarchical information in parameters passed to PL/SQL functions and procedures. The Microsoft BizTalk Adapter for Oracle Database surfaces RECORD types as complex XML types. For more information about how the Oracle Database adapter supports RECORD Types, see [Operations on Functions and Procedures with RECORD Types](#). For information about the XML structure for RECORD Types, see [Message Schemas for RECORD Types](#).

How to Invoke Functions in an Oracle Database?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a function in an Oracle database that returns simple and nested RECORD types, these tasks are:

1. Create a BizTalk project and generate schema for the function you want to invoke in an Oracle database.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the function in the Oracle database.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `Func_RecordTypes`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how the Oracle Database adapter supports invoking function that return RECORD type parameters we will invoke:

- The `GET_ACCOUNTADDRESS` function that returns a simple RECORD type.
- The `GET_ACCOUNTINFO` function that returns a nested RECORD type.

These functions are available as part of the `ACCOUNT_PKG` created by running the SQL scripts provided with the samples. To know more about the samples and the SQL scripts, see [Samples](#).

So, we must generate schema for the both the functions, `GET_ACCOUNTADDRESS` and `GET_ACCOUNTINFO`, available under the `SCOTT\Package\ACCOUNT_PKG` schema. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two request-response message sets—one request-response set to invoke the `GET_ACCOUNTADDRESS` function and receive a response; the other request-response message set to invoke the `GET_ACCOUNTINFO` function and receive a response.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message, and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>Func_RecordTypes.OracleDBBindingSchema.GET_ACCOUNTINFO</i> , where <i>Func_RecordTypes</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the GET_ACCOUNTADDRESS function.

5. Repeat the previous step to create three more messages. In the **Properties** pane for the new messages, do the following:

Set Identifier to	Set Message Type to
Response	<i>Func_RecordTypes.OracleDBBindingSchema.GET_ACCOUNTINFOResponse</i>
Request2	<i>Func_RecordTypes.OracleDBBindingSchema.GET_ACCOUNTADDRESS</i>
Response2	<i>Func_RecordTypes.OracleDBBindingSchema.GET_ACCOUNTADDRESSResponse</i>

Setting up the Orchestration

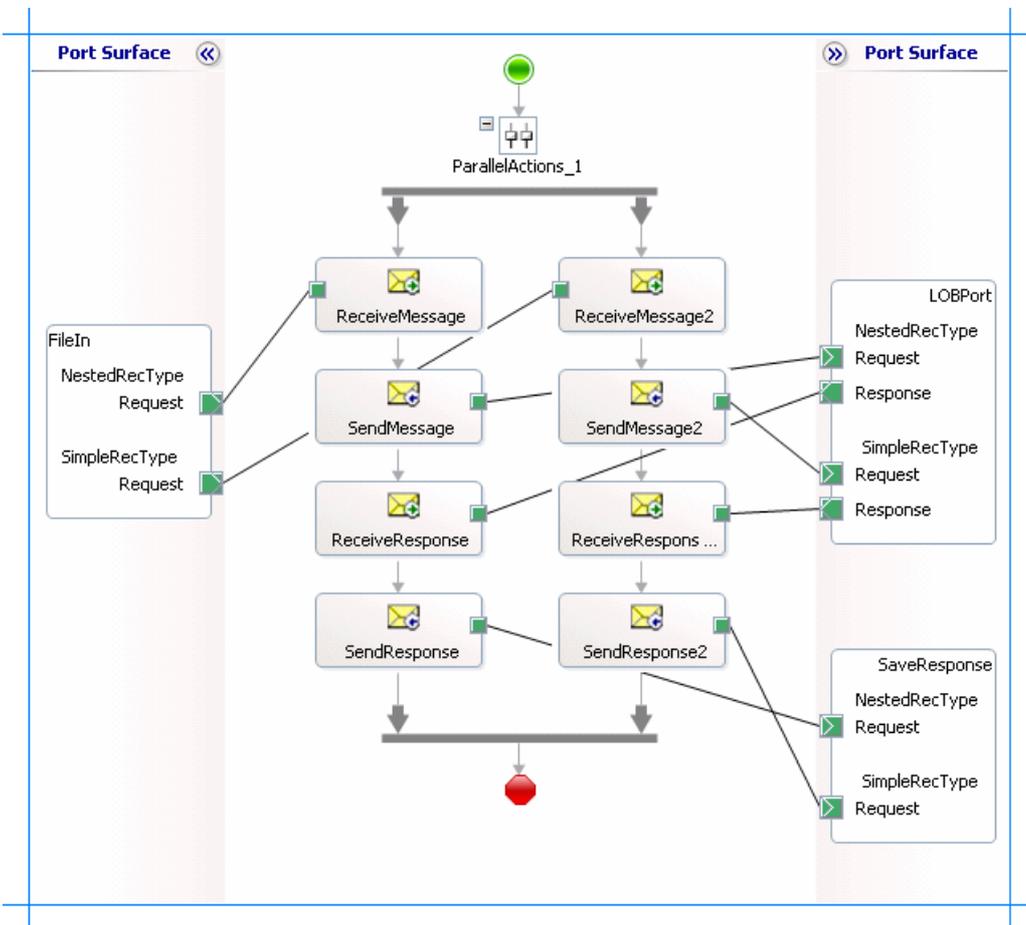
You must create a BizTalk orchestration to use BizTalk Server for invoking a function returns simple and complex RECORD types. In this orchestration, you drop two request messages:

- One for the GET_ACCOUNTADDRESS function that returns a simple RECORD type.
- One for the GET_ACCOUNTINFO function that returns a nested RECORD type.

These messages are dropped at a receive location. The Oracle Database adapter consumes the messages and passes them on to the Oracle database via ODP. The response from the Oracle database is saved to another location. Because the orchestration picks two requests simultaneously, you need to include a Parallel Actions shape in the orchestration. For each parallel action, you must include Send and Receive shapes to send messages to Oracle database and receive responses, respectively. However, you could use the same ports for sending and receiving messages for both operations. A typical orchestration for performing both operations simultaneously would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration. The following table lists the shapes you must include for one of the parallel actions.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

The following table lists the shapes that you must include for the other parallel action.

Shape	Shape Type	Properties
ReceiveMessage2	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage2</i> Set Activate to <i>True</i>
SendMessage2	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage2</i>

ReceiveResponse2	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse2</i> • Set Activate to <i>False</i>
SendResponse2	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse2</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Because you will be processing two request and response messages using these ports, you must create two operations for each port, where each operation corresponds to one message type. To create an operation, right-click the port shape and select **New Operation**. Name the first operation for each port as **NestedRecType** and the second operation for each port as **SimpleRecType**.

Using Correlations

Correlation is the process of matching an incoming message with the appropriate instance of an orchestration. In the orchestration you will be dropping two request messages, one for each overload. Using correlation, you associate a request message with the right orchestration. For more information about correlation, see <http://go.microsoft.com/fwlink/?LinkId=103055>.

To use correlations

1. Promote a property from the schema generated for each function. For example, promote the CUSTNAME property from the schema of GET_ACCOUNTADDRESS function; promote the AID property from the schema of GET_ACCOUNTINFO function. To promote a property, right-click the property in the schema view, point to **Promote**, and then select **Quick Promotion**. This adds a PropertySchema.xsd file to your BizTalk project.

For information about promoting a property, see <http://go.microsoft.com/fwlink/?LinkId=102358>.

2. From Orchestration View, right-click **Correlation Types**, and then select **New Correlation Type**.
3. The **Correlation Properties** dialog box lists the properties you promoted in step 1. Select a property, and then click **Add**.
4. Click **OK**.
5. To create correlation types for the other promoted property, repeat these steps.
6. Rename the correlation types based on the property to which they are associated. You could rename the correlation types to *CorrelationType_CUSTNAME* (for CUSTNAME property) and *CorrelationType_AID* (for AID property).
7. From Orchestration View, right-click **Correlation Sets**, and then select **New Correlation Set**.
8. Right-click the newly added correlation set, and then click **Properties**. In the **Properties** pane, do the following:

Use this	To do this
Correlation Type	<i>Func_RecordTypes.CorrelationType_CUSTNAME</i>
Identifier	<i>Correlation_CUSTNAME</i>

9. Add another correlation set and specify the following properties from the **Properties** pane.

Use this	To do this
Correlation Type	<i>Func_RecordTypes.CorrelationType_AID</i>
Identifier	<i>Correlation_AID</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link them to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_AID</i> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.NestedRecType.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.NestedRecType.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.NestedRecType.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.NestedRecType.Request</i>

ReceiveMessage2	<ul style="list-style-type: none"> • Set Initializing Correlation Sets to <i>Correlation_CUSTNAME</i> • Set Message to <i>Request2</i> • Set Operation to <i>FileIn.SimpleRecType.Request</i>
SendMessage2	<ul style="list-style-type: none"> • Set Message to <i>Request2</i> • Set Operation to <i>LOBPort.SimpleRecType.Request</i>
ReceiveResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response2</i> • Set Operation to <i>LOBPort.SimpleRecType.Response</i>
SendResponse2	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.SimpleRecType.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop the request messages, one for each overloaded procedure. The BizTalk orchestration will consume the request messages and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response messages, one for each function, containing the response from the Oracle database.
- Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Because the WCF-Custom or WCF-OracleDB send port sends and receive messages conforming to more than one schema and performs two operations, you must set dynamic action for both the operations. For more information about actions, see [Specifying SOAP Action](#). For this orchestration, the action should be set as follows:

```
<BtsActionMapping>
  <Operation Name="NestedRecType" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTINFO" />
  <Operation Name="SimpleRecType" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS" />
</BtsActionMapping>
```

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking a function in an Oracle database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop two request messages (one for each function) to the FILE receive location. The schema for the request messages must conform to the schema for the functions you generated earlier. The orchestration consumes the request messages and sends them to the Oracle database. The response from the Oracle database is saved at other FILE location defined as part of the orchestration.

See [Message Schemas for Functions and Procedures](#) for more information about the request message schema for invoking functions using the Oracle Database adapter.

For example, the request message to invoke the GET_ACCOUNTINFO function is:

```
<GET_ACCOUNTINFO xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <AID>100000</AID>
</GET_ACCOUNTINFO>
```

Similarly, the request message to invoke the GET_ACCOUNTADDRESS function is:

```
<GET_ACCOUNTADDRESS xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <CUSTNAME>Mindy Martin</CUSTNAME>
</GET_ACCOUNTADDRESS>
```

The first request message invokes the GET_ACCOUNTINFO function that returns a nested record type. The response message for invoking the GET_ACCOUNTINFO function is:

```
<?xml version="1.0" encoding="utf-8" ?>
<GET_ACCOUNTINFOResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <GET_ACCOUNTINFOResult>
    <ACCT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTINFO">
      <ACCTID>100000</ACCTID>
      <NAME>Kim Ralls</NAME>
      <BALANCE>10000</BALANCE>
    </ACCT>
    <ADDRESS xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTINFO">
```

```
<STREET>1234 Main St.</STREET>
<CITY>Seattle</CITY>
<STATE>WA</STATE>
</ADDRESS>
</GET_ACCOUNTINFOResult>
</GET_ACCOUNTINFOResponse>
```

The second request message invokes the GET_ACCOUNTADDRESS function that returns a simple record type. The response message for invoking the GET_ACCOUNTADDRESS function is:

```
<?xml version="1.0" encoding="utf-8"?>
<GET_ACCOUNTADDRESSResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
  <GET_ACCOUNTADDRESSResult>
    <ID xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS">100004</ID>
    <NAME xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS">Mindy Martin</NAME>
    <STREET xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS">6789 Cherry St.</STREET>
    <CITY xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS">New York</CITY>
    <STATE xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTADDRESS">NY</STATE>
  </GET_ACCOUNTADDRESSResult>
</GET_ACCOUNTADDRESSResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while invoking functions or procedures using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Tables with BFILE Data Types by Using BizTalk Server

The Oracle Database adapter supports the BFILE data type in tables and stored procedures. This section provides information on how to perform operations on tables that have a column of BFILE data type. For more information about how the Oracle Database adapter supports BFILE, see [Operations on Tables With BFILE Data Types](#).

Setting Up Your Oracle Database Server for Operations on BFILE

This section demonstrates how to invoke a procedure that inserts a record into the SCOTT.CUSTOMERDOC table. This table contains a column of BFILE data type and is created by running the SQL scripts shipped with the BizTalk Adapter Pack samples. To know more about the samples and the SQL scripts, see [Samples](#).

After you have run the script to create the CUSTOMERDOC table, you must perform certain actions on the computer running the Oracle database to enable operations on BFILE data types. The tasks that you must perform on the Oracle database are:

1. Create a directory C:\MYDIR on the computer running the Oracle database.
2. Create a logical directory in the Oracle database. This usually requires a user with SYSDBA privileges. For example:

```
CREATE OR REPLACE DIRECTORY MYDIR AS 'C:\MYDIR';
```

3. Add privileges to the user to access the logical directory in Oracle. For example:

```
GRANT READ, WRITE ON DIRECTORY MYDIR to SCOTT;
```

4. Copy the files to be accessed into the physical directory location, on the computer running the Oracle database, associated with the logical directory in Oracle. You created this directory in step 1.

Based on the example above, copy a file, customer_profile.txt into directory C:\MYDIR. This file is now available for BFILE operations. For more information about performing operations, see [Performing Operations on Tables with Large Object Types Data by Using BizTalk Server](#).

◆ Important

The ReadLOB operation is supported on tables with the BFILE data type. The UpdateLOB operation is NOT supported. However, users can alternately use the UPDATE operation.

How to Perform Operations Using BFILE Data Types

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a procedure that inserts a record into the SCOTT.CUSTOMERDOC table, these tasks are:

1. Create a BizTalk project and generate schema for the CREATE_CUSTOMERDOC stored procedure.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the operation on the Oracle database table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, Operate_BFILE, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to perform operations on a table with BFILE columns, we will invoke the CREATE_CUSTOMERDOC procedure. This procedure is created under the SCOTT\Package\ACCOUNT_PKG schema by running the SQL scripts provided with the samples. This procedure takes BFILE record type and adds a record in the CUSTOMERDOC table. For more information about the SQL scripts, see [Samples](#).

See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Oracle database and the other to receive a response.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and then select <i>BFILE_Operations.OracleDBBindingSchema.CREATE_CUSTOMERDOC</i> , where <i>BFILE_Operations</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the CREATE_CUSTOMERDOC procedure.

5. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and then select <i>BFILE_Operations.OracleDBBindingSchema.CREATE_CUSTOMERDOCResponse</i> .

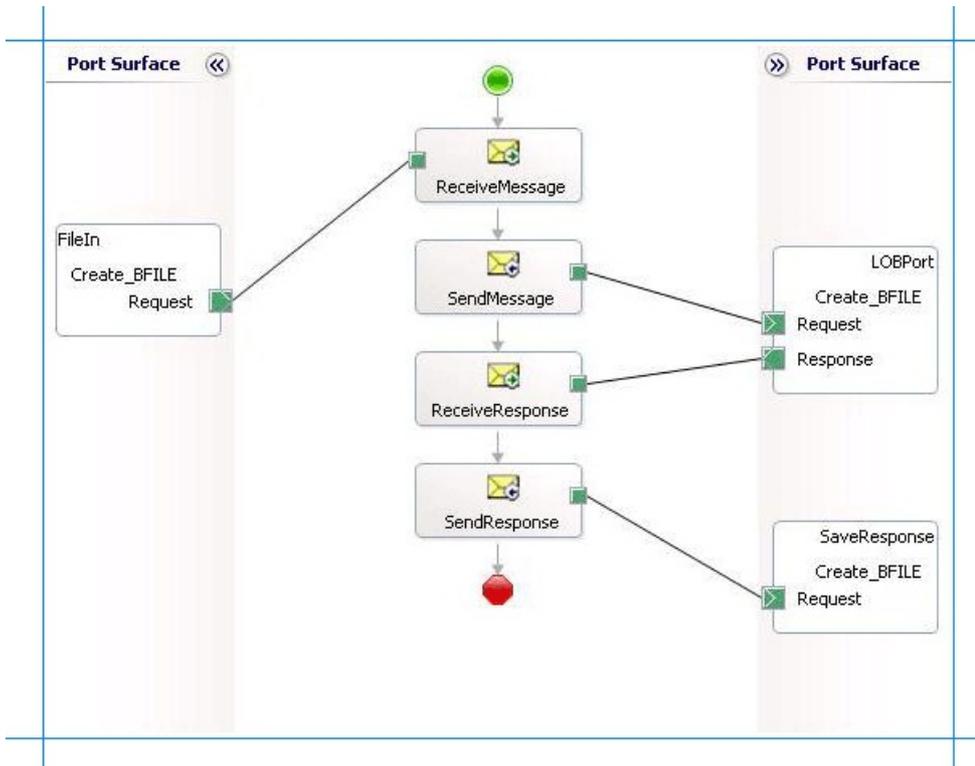
Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for executing a procedure. In this orchestration, you drop a request message at a defined receive location. The Oracle Database adapter consumes this message and passes it on to the Oracle database via ODP. The response from the Oracle database is saved to another location. A typical orchestration for performing operations on BFILE columns in an Oracle database table would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.

- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
------	------------

FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Create_BFILE.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Create_BFILE.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Create_BFILE.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Create_BFILE.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking the procedure that creates a record in the CUSTOMERDOC table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the procedure you generated earlier. See [Message Schemas for Functions and Procedures](#) for more information about the request message schema for invoking procedure using the Oracle Database adapter.

For example, the request message to invoke the CREATE_CUSTOMERDOC procedure is:

```
<CREATE_CUSTOMERDOC xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCO
UNT_PKG">
  <CNAME>John Smith</CNAME>
  <CDOC>MYDIR/John_Smith_profile.txt</CDOC>
</CREATE_CUSTOMERDOC>
```

Note

The text file, John_Smith_profile.txt must be present in the physical directory location associated with the logical directory in Oracle. For this example, the text file must be present in C:\MYDIR

The orchestration consumes the message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the above request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<CREATE_CUSTOMERDOCResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Pack
age/ACCOUNT_PKG"></CREATE_CUSTOMERDOCResponse>
```

Note

You can create a similar orchestration to read data from tables that have BFILE type fields. The SQL script shipped with the BizTalk Adapter Pack creates an ACCOUNT_PKG that contains a GET_CUSTOMERDOC procedure. You can use this procedure to retrieve BFILE data from the SCOTT.CUSTOMERDOC table.

A sample, Operate_BFILE, is also included with the samples for BizTalk Adapter Pack. This sample demonstrates how to insert records into the SCOTT.CUSTOMERDOC table using the CREATE_CUSTOMERDOC stored procedure (as described in this topic.) The sample also demonstrates how to read BFILE data from the SCOTT.CUSTOMERDOC table using GET_CUSTOMERDOC stored procedure.

Possible Exceptions

For information about the exceptions you might encounter while performing a DML operation using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing a SQLEXECUTE Operation by Using BizTalk Server

The Oracle Database adapter enables clients to run parameterized SQL statement on an Oracle database. To support such operations, the Oracle Database adapter surfaces a SQLEXECUTE operation. SQLEXECUTE operation supports an input parameter block comprised of parameter sets that enable you to execute the same SQL statement once for each set. The SQLEXECUTE operation returns the results of the SQL statement in a generic record set. For more information about the operation, see [SQLEXECUTE Operation](#). For information about the structure of the SOAP message for SQLEXECUTE operation, see [Message Schemas for the SQLEXECUTE Operation](#).

How to Perform a SQLEXECUTE operation on an Oracle Database?

Performing an operation on an Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform a SQLEXECUTE operation, these tasks are:

1. Create a BizTalk project and generate schema for the SQLEXECUTE operation. The SQLEXECUTE operation is surfaced under the root node (/) in the **Select a category** pane in the Consume Adapter Service BizTalk Project Add-in.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle database.
3. Create an orchestration to invoke the operation on the Oracle database table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, SqlExec, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to run a parameterized SQL query, we will generate schema for SQLEXECUTE operation available at the root node (/) in the **Select a category** pane in the Consume Adapter Service Add-in. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration View window of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Oracle database and the other to receive a response.

Perform the following steps to create messages and link them to the schema.

To create messages and link to schema

1. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
2. In Orchestration View, right-click **Messages**, and then click **New Message**.
3. Right-click the newly created message and then select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
his	

Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and then select <i>SqlExec.OracleDBBindingSchema.SQLEXECUTE</i> , where <i>SqlExec</i> is the name of your BizTalk project. <i>OracleDBBindingSchema</i> is the schema generated for the SQLEXECUTE operation.

5. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

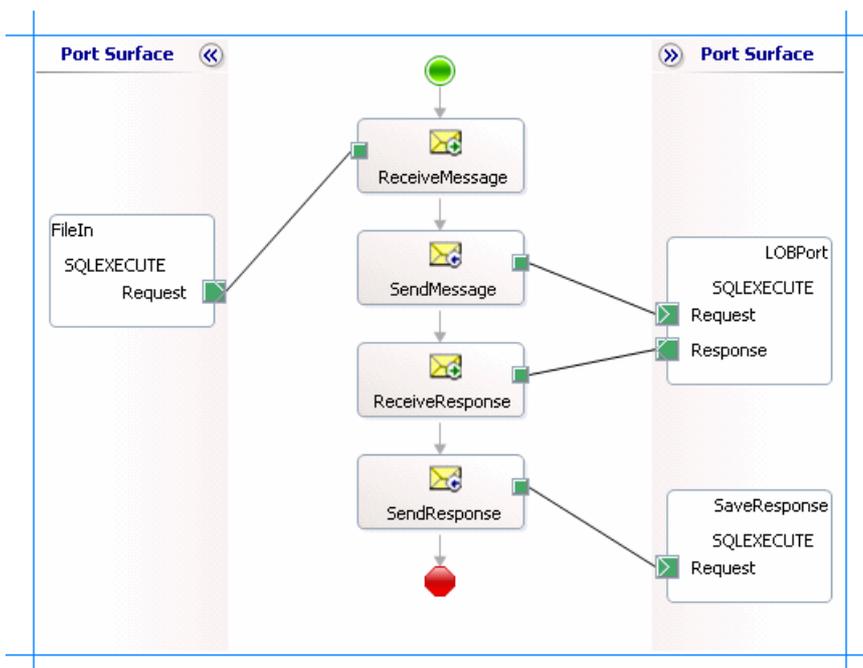
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and then select <i>SqlExec.OracleDBBindingSchema.SQLEXECUTE Response</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for running a parameterized SQL query using the SQLEXECUTE operation. In this orchestration, you drop a request message at a defined receive location. The Oracle Database adapter consumes this message and passes it on to the Oracle database via ODP. The response from the Oracle database is saved to another location. A typical orchestration for performing SQLEXECUTE operation on Oracle database would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration for the SQLEXECUTE operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.SQLEXECUTE.Request</i>

SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.SQLEXECUTE.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.SQLEXECUTE.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.SQLEXECUTE.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create WCF-Custom or WCF-OracleDB ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for performing the SQLEXECUTE operation. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.

- The WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the SQLEXECUTE operation you generated earlier. The orchestration consumes the message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration. See [Message Schemas for the SQLEXECUTE Operation](#) for more information about the request message schema for invoking a SQLEXECUTE operation.

Because the SQLEXECUTE operation is not surfaced under any Oracle database artifact, you can use the same schema to perform a parameterized SQL query on a view or execute a procedure that operates on some other table.

For example, the following request message performs a parameterized SELECT statement on the ACCOUNT table using the SQLEXECUTE operation. The ACCOUNT table is created under the SCOTT schema by running the SQL scripts provided with the samples. To know more about the samples, see [Samples](#).

```
<SQLEXECUTE xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE">
  <SQLSTATEMENT>select * from ACCOUNT where ACCTID=:num</SQLSTATEMENT>
  <PARAMETERSCHEMA>num number</PARAMETERSCHEMA>
  <PARAMETERSET>
    <PARAMETERDATA xmlns="http://Microsoft.LobServices.OracleDB/2007/03">
      <PARAMETER>
        <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">100000</s
tring>
      </PARAMETER>
    </PARAMETERDATA>
  </PARAMETERSET>
</SQLEXECUTE>
```

The response from Oracle database for the above request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<SQLEXECUTEResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE">
  <SQLEXECUTEResult>
    <GenRecordRow xmlns="http://Microsoft.LobServices.OracleDB/2007/03">
      <GenRecordColumn>
        <GenRecordColumn>
          <ColumnName>ACCTID</ColumnName>
          <ColumnValue>100000</ColumnValue>
          <ColumnType>System.Decimal</ColumnType>
        </GenRecordColumn>
        <GenRecordColumn>
          <ColumnName>NAME</ColumnName>
          <ColumnValue>Kim Ralls</ColumnValue>
          <ColumnType>System.String</ColumnType>
        </GenRecordColumn>
        <GenRecordColumn>
          <ColumnName>BALANCE</ColumnName>
          <ColumnValue>10000</ColumnValue>
          <ColumnType>System.Decimal</ColumnType>
        </GenRecordColumn>
      </GenRecordRow>
    </SQLEXECUTEResult>
  </SQLEXECUTEResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while performing a DML operation using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Polling Oracle Database by Using BizTalk Server

You can configure the Oracle Database adapter to receive polling-based messages from Oracle database. The adapter provides two ways of polling the Oracle database:

- **Using SELECT statements.** You can specify a simple SELECT statement to poll the tables and views in the Oracle database. The adapter executes the SELECT statement at specified intervals and returns the result to the adapter clients.
- **Using stored procedures, functions, or procedures or functions within a package.** You can specify a stored procedure, function, or procedure or function within a package to poll the Oracle database. The adapter executes the request message at specified intervals and returns the result to the adapter clients.

The key difference in the two approaches is the way adapter clients specify a polling statement that the adapter uses to poll the Oracle database. While the polling statement for the first approach is a simple SELECT statement, the polling statement for the other approach is a request message that executes the stored procedure, function, or procedure or function within a package. Adapter clients specify the polling statement, for either approach, in the **PollingStatement** binding property. For more information about the binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

The topics in this section provide instructions on how to poll using a SELECT statement and a stored procedure, function, or procedure or function within a package.

In This Section

- [Polling Oracle Database Using the SELECT Statement](#)
- [Polling Oracle Database Using Stored Procedures, Functions, or Packaged Procedures and Functions](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Polling Oracle Database Using the SELECT Statement

You can configure the Oracle Database adapter to receive periodic data-change messages by using a SELECT statement to continuously poll the tables and views in Oracle the Oracle database. You can specify a SELECT statement as a polling statement that the adapter executes periodically to poll the Oracle database. Optionally, you can also specify a post-poll PL/SQL code block that the adapter executes if there is a change in data. This block is often used to update a field on the queried records in the target or to move the queried records to another table or view.

To enable this, you must specify certain binding properties on the Oracle Database adapter. You can also modify the target namespace for the POLLINGSTMT operation by setting the **PollingId** property in the connection URI. For more information, see [Support for Receiving Polling-based Data-changed Messages](#) and [Receiving Polling-based Data-changed Messages](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling Operations](#).

Configuring a Polling Operation with Oracle Database Adapter Binding Properties

The following table summarizes the Oracle Database adapter binding properties that you use to configure the adapter to receive data change messages. You must specify these binding properties while configuring the receive port in BizTalk Server Administration console.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the SELECT statement you specify for the PollingStatement binding property will be executed. The default is <code>SELECT 1 FROM DUAL</code> , which implies that the adapter must continue polling irrespective of whether the table being polled has data or not.
PollingInterval	Specifies the interval, in seconds, at which the Oracle Database adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 500 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingStatement	Specifies the polling statement. To poll using a SELECT statement, you must specify a SELECT statement for this binding property. The default is null. You must specify a value for PollingStatement binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingStatement binding property is executed.
PollWhileDataFound	Specifies whether the Oracle Database adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For a complete description of how to use the Oracle Database adapter to poll the Oracle database, read further.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle Database adapter supports receiving data change messages using SELECT statements, create a BizTalk project and generate schema for the **POLLINGSTMT** operation by setting the **PollingStatement**

binding property to the following:

```
SELECT * FROM ACCOUNTACTIVITY FOR UPDATE
```

The ACCOUNTACTIVITY table is created when you run the SQL scripts provided with the samples to create these objects in the database.

Note

The orchestration in this topic polls the ACCOUNTACTIVITY table, which is a base database table created by running the scripts provided with the samples. You must perform similar procedures as described in this topic to poll any other table.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the table being polled (ACCOUNTACTIVITY) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM ACCOUNTACTIVITY
```

This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.

- Specify the SELECT statement as stated earlier for the **PollingStatement** binding property. This statement retrieves all the rows in the ACCOUNTACTIVITY table.
- EXECUTE a PL/SQL block as part of the **PostPollStatement** binding property. This statement will move all data from ACCOUNTACTIVITY table to another table in the database. Once this happens, the next time the statement specified for **PollingStatement** will be executed, it will not fetch any data.
- Until more data is added to the ACCOUNTACTIVITY table, you will not get any polling messages. So, you must repopulate the ACCOUNTACTIVITY table with new records. You can do so by running the more_activity_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

How to Receive Data-change Messages from Oracle

Performing an operation on Oracle database using Oracle Database adapter with BizTalk Server involves the following procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to poll Oracle database using a SELECT statement, these tasks are:

1. Create a BizTalk project, and generate schema for the **POLLINGSTMT** operation for the table you want to poll.
2. Create a message in the BizTalk project for receiving messages from Oracle database.
3. Create an orchestration to receive messages from Oracle and save them to a folder.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **POLLINGSTMT** operation. Perform the following tasks while generating the schema using the Consume Adapter Service Add-in.

- Specify a value for **PollingStatement** binding property while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For example, specify the following as a polling statement:

```
SELECT * FROM ACCOUNTACTIVITY FOR UPDATE
```

- Select the contract type as **Service (Inbound operation)**.
- Generate schema for the **POLLINGSTMT** operation.

For more information about how to generate schema, see [Browsing, Searching, and Retrieving Metadata for Oracle Operations](#).

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from Oracle.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

U	To do this
s	
e	
t	
h	
i	
s	
I	Type Receive .
d	
e	
n	
t	
i	
f	
i	
e	

From the drop-down list, expand **Schemas**, and select *TablePolling.OracleDBBinding*, where *TablePolling* is the name of your BizTalk project. *OracleDBBindingSchema* is the response schema generated for the **POLLINGSTMT** operation on ACCOUNTACTIVITY table.

Important

Because polling is a one way operation, the schema generated by the adapter does not contain a response node, and hence there is only one root node in the schema. If you use such schemas for a message type, you must identify the schema by the filename of the generated schema. For example, if you create schema for a two-way operation, the nodes in the schema file with a name *OracleDBBindingSchema* may look like "Request" and "Response". If you want to create a message in the orchestration that maps to the request schema, you can identify the schema in the list by looking for *OracleDBBindingSchema.Request*. However, in the case of polling operation, because the only node is "POLLINGSTMT", it is not easy to identify the schema you want to map to because schemas with single nodes are not listed as <schemafilename>.<rootnodename>. Instead, such schemas are listed by only the filename. In such a case, the only way to identify the schema is by the schema filename, for example, *OracleDBBindingSchema*.

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from Oracle. In this orchestration, the adapter receives the response by executing the SELECT statement specified for the **PollingStatement** binding property. The response message for the SELECT statement is saved to a FILE location. A typical orchestration for polling Oracle database would contain:

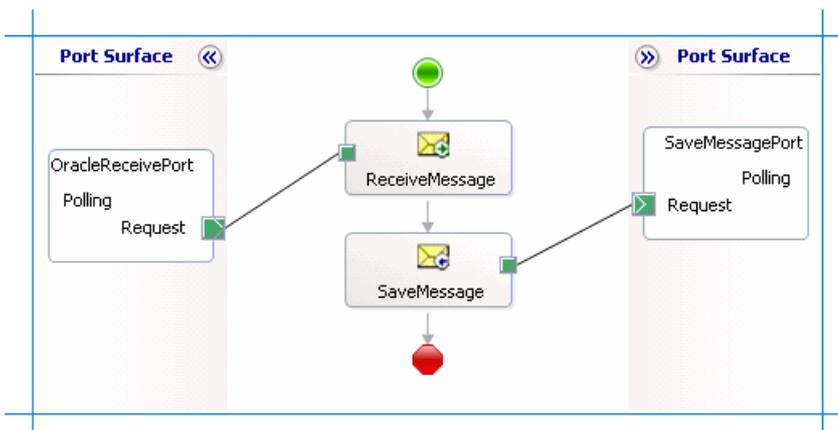
- Receive and Send shapes to receive messages from Oracle and send to a FILE port, respectively.
- A one-way receive port to receive messages from Oracle database.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from Oracle database.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>

SaveMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveMessage</i>
-------------	------	---

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleReceivePort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleReceivePort</i> • Set Type to <i>OracleReceivePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>OracleReceivePort.Polling.Request</i>
SaveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SaveMessagePort.Polling.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:

- Define a location on the hard disk and a corresponding FILE port where the BizTalk orchestration will drop the messages from Oracle. These messages will be in response to the polling statement that you specify for the receive port.
- Define a physical WCF-Custom or WCF-OracleDB one-way receive port. This port polls the Oracle database. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Make sure you specify the following binding properties for the receive port.

Binding Property	Value
InboundOperationType	Set this to Polling .
PolledDataAvailableStatement	<p>For this example, set this binding property to:</p> <pre>SELECT COUNT (*) FROM ACCOUNTACTIVITY</pre> <p>This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.</p>
PollingStatement	<p>For this binding property, specify a SELECT statement to retrieve all records from ACCOUNTACTIVITY table. For this example, set this binding property to:</p> <pre>SELECT * FROM ACCOUNTACTIVITY FOR UPDATE</pre>
PostPollStatement	<p>Specify the post-poll statement to move all data from ACCOUNTACTIVITY table to another table. For this example, set this binding property to:</p> <pre>BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;</pre>

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

 Note
We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle Database adapter. You can do so by adding the service behavior while configuring the receive port. For instruction on how to add the service behavior, see Configure Transaction Isolation Level and Transaction Timeout .

Starting the Application

You must start the BizTalk application for polling Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleDB one-way receive port, which polls Oracle using the SELECT statement specified for the **PollingStatement** binding property, is running.
- The FILE send port, which receives messages from Oracle database, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** which returns a positive value indicating the adapter to execute the statement specified for **PollingStatement** binding property.
- The adapter executes the SELECT statement for the **PollingStatement** binding property and returns all the rows in the ACCOUNTACTIVITY table. The response from Oracle database resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<POLLINGSTMT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMT">
  <POLLINGSTMTRECORD>
    <POLLINGSTMTRECORD>
      <TID>1</TID>
      <ACCOUNT>100001</ACCOUNT>
      <AMOUNT>500</AMOUNT>
      <DESCRIPTION />
      <TRANSDATE>2008-08-03T20:10:28</TRANSDATE>
      <PROCESSED>n</PROCESSED>
    </POLLINGSTMTRECORD>
  </POLLINGSTMTRECORD>
  .....
  .....
</POLLINGSTMTRECORD>
  .....
  .....
</POLLINGSTMTRECORD>
</POLLINGSTMT>
```

- The adapter executes the post-poll statement, which moves all the data from ACCOUNTACTIVITY table to another table.
- After the polling interval, the adapter again executes **PolledDataAvailableStatement**. Because ACCOUNTACTIVITY table has no records now, **PolledDataAvailableStatement** does not return a positive value and hence the adapter does not execute the statement specified for the **PollingStatement** binding property. As a result, adapter client does not get any polling message.
- The adapter client will not get any more polling messages until some records are explicitly inserted into the ACCOUNTACTIVITY table. To insert more records, you can run the more_activity_data.sql script provided with the samples. After you run this script, the next time **PolledDataAvailableStatement** is executed, it returns a positive value. As a result, the adapter executes the polling statement and adapter clients again receive a polling message.

Note

The Oracle Database adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Possible Exceptions

For information about the exceptions you might encounter while running a polling query on the Oracle database using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling Oracle Database by Using BizTalk Server](#)

Polling Oracle Database Using Stored Procedures, Functions, or Packaged Procedures and Functions

You can configure the Oracle Database adapter to receive periodic data-change messages by using stored procedures, functions, or packaged procedures and functions to periodically poll the Oracle database. You can specify a stored procedure, function, or packaged procedure and function as a polling statement that the adapter executes periodically to poll the Oracle database.

To enable this, you must specify certain binding properties on the Oracle Database adapter. You can also modify the target namespace for the POLLINGSTMT operation by setting the **PollingId** property in the connection URI. For more information, see [Support for Receiving Polling-based Data-changed Messages](#) and [Receiving Polling-based Data-changed Messages](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling Operations](#).

Configuring a Polling Operation with Oracle Database Adapter Binding Properties

The Oracle Database adapter executes the polling query and any post-poll PL/SQL code block inside a transaction. The following table summarizes the Oracle Database adapter binding properties that you use to configure the adapter to receive data change messages. You must specify these binding properties while configuring the WCF-Custom or WCF-OracleDB receive port in the BizTalk Server Administration console.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform the Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the stored procedure you specified for the PollingStatement binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle Database adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 500 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval. The default is <code>SELECT 1 FROM DUAL</code> , which implies that the adapter must continue polling irrespective of whether the table being polled has data or not.
PollingStatement	Specifies the polling statement. To poll using a stored procedure, function, or packaged procedure or function, you must specify the entire request message for the respective operation in this binding property. The request message must be the same that you send to the adapter for invoking the respective operations as an outbound operation. The default is null. You must specify a value for PollingStatement binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingStatement binding property is executed.

PollWhileDataFound	Specifies whether the Oracle Database adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.
---------------------------	---

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For a complete description of how to use the Oracle Database adapter to poll the Oracle database, read further.

How This Topic Demonstrates Polling

This topic demonstrates how you can poll the Oracle database using a stored procedure. Create a BizTalk project and generate schema for the stored procedure you want to use to poll the Oracle database. In this topic, we use the GET_ACTIVITYYS stored procedure to poll the ACCOUNTACTIVITY table. This stored procedure is available with the ACCOUNT_PKG package in the SCOTT schema. You can run the SQL scripts provided with the samples to create these objects in the database.

Note
The orchestration in this topic polls the ACCOUNTACTIVITY table, which is a database table created by running the scripts provided with the samples. You must perform similar procedures as described in this topic to poll any other table.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the table being polled (ACCOUNTACTIVITY) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM ACCOUNTACTIVITY
```

This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.

- Execute a stored procedure, GET_ACTIVITYYS, by providing the request message as part of the **PollingStatement** binding property. This stored procedure will retrieve all the rows in the ACCOUNTACTIVITY table and you will get a response message from the adapter.
- EXECUTE a PL/SQL block as part of the **PostPollStatement** binding property. This statement will move all data from ACCOUNTACTIVITY table to another table in the database. Once this happens, the next time **PollingStatement** will be executed, it will not fetch any data and hence the GET_ACTIVITYYS stored procedure will return an empty response message.
- Until more data is added to the ACCOUNTACTIVITY table, you will continue to get empty response messages. So, you must repopulate the ACCOUNTACTIVITY table with new records. You can do so by running the more_activity_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

How to Receive Data-change Messages from Oracle

Performing an operation on Oracle database using Oracle Database adapter with BizTalk Server involves the following procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to poll Oracle database using a stored procedure:

1. Create a BizTalk project, and generate schema for the stored procedure you want to use for polling.
2. Create a message in the BizTalk project for receiving messages from Oracle database.
3. Create an orchestration to receive messages from Oracle database and save them to a folder.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the GET_ACTIVITYYS operation. Perform the following tasks while generating the schema using the Consume Adapter Service Add-in.

- Select the contract type as **Service (Inbound operation)**.
- Generate schema for the **GET_ACTIVITYYS** procedure.

For more information about how to generate schema, see [Browsing, Searching, and Retrieving Metadata for Oracle Operations](#).

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from Oracle.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

U s e t h is	To do this
I d e n t i f i e r	Type Receive .

From the drop-down list, expand **Schemas**, and select *Polling.OracleEBSBindingSchema*, where *Polling* is the name of your BizTalk project. *OracleEBSBindingSchema* is the response schema generated for the **GET_ACTIVITYS** stored procedure.

Important

Because polling is a one way operation, the schema generated by the adapter does not contain a response node, and hence there is only one root node in the schema. If you use such schemas for a message type, you must identify the schema by the filename of the generated schema. For example, if you create schema for a two-way operation, the nodes in the schema file with a name *OracleEBSBindingSchema* may look like "Request" and "Response". If you want to create a message in the orchestration that maps to the request schema, you can identify the schema in the list by looking for *OracleEBSBindingSchema.Request*. However, in the case of polling operation, because the only node is "Poll", it is not easy to identify the schema you want to map to because schemas with single nodes are not listed as <schemafilename>.<rootnodename>. Instead, such schemas are listed by only the filename. In such a case, the only way to identify the schema is by the schema filename, for example, *OracleEBSBindingSchema*.

The Consume Adapter Service Add-in generates schema for both inbound and outbound operations for the GET_ACTIVITYS stored procedure. You must use the schema for inbound operation to:

- Map the message created as part of the orchestration.
- To retrieve the action you must specify for the **PollingAction** binding property at run-time.

You must use the schema for the outbound operation to get the request message you must specify as part of the **PollingStatement** binding property.

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from Oracle. In this orchestration, the adapter receives the response by executing the stored procedure for which you specified the request message as part of the **PollingStatement** binding property. The response message for the stored procedure is saved to a FILE location. A typical orchestration for polling Oracle database would contain:

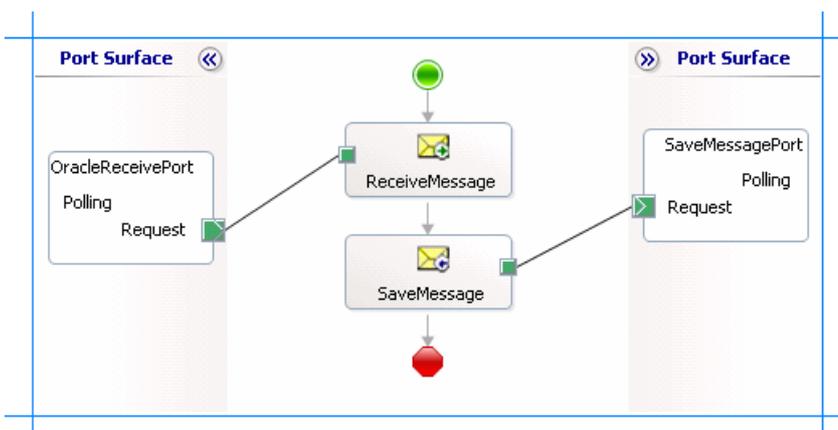
- Receive and Send shapes to receive messages from Oracle and send to a FILE port, respectively.
- A one-way receive port to receive messages from Oracle database.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from Oracle database.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SaveMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SaveMessage</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleReceivePort	<ul style="list-style-type: none"> Set Identifier to <i>OracleReceivePort</i> Set Type to <i>OracleReceivePortType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>
SaveMessagePort	<ul style="list-style-type: none"> Set Identifier to <i>SaveMessagePort</i> Set Type to <i>SaveMessagePortType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> Set Message to <i>Receive</i> Set Operation to <i>OracleReceivePort.Polling.Request</i>
SaveMessage	<ul style="list-style-type: none"> Set Message to <i>Receive</i> Set Operation to <i>SaveMessagePort.Polling.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application.

For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding FILE port where the BizTalk orchestration will drop the messages from Oracle. These messages will be in response to the polling statement that you specify for the receive port.
 - Define a physical WCF-Custom or WCF-OracleDB one-way receive port. This port polls the Oracle database. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Make sure you specify the following binding properties for the receive port.

Binding Property	Value
Inbound OperationType	Set this to Polling .
PolledDataAvailableStatement	<p>For this example, set this binding property to:</p> <pre>SELECT COUNT (*) FROM ACCOUNTACTIVITY</pre> <p>This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.</p>
Polling Action	Retrieve the polling action from the schema generated for the inbound message for the GET_ACTIVITY S procedure. For this example, set this binding property to http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/PollingPackage/ /ACCOUNT_PKG/GET_ACTIVITIES .
PollingStatement	<p>For this binding property, specify the request message to invoke the GET_ACTIVITY S stored procedure. You can get the request message from the schema for the outbound operation generated by the Consume Adapter Service Add-in. You must provide the entire XML message as an input for this binding property. For this example, set this binding property to:</p> <pre><GET_ACTIVITIES xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACTIVITY"> <INRECS>OPEN ? FOR SELECT * FROM ACCOUNTACTIVITY</INRECS> </GET_ACTIVITIES></pre> <p>The GET_ACTIVITY S stored procedure takes an input REF CURSOR as parameter.</p>
PostPollStatement	<p>Specify the post-poll statement to move all data from ACCOUNTACTIVITY table to another table. For this example, set this binding property to:</p> <pre>BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;</pre>

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle Database adapter. You can do so by adding the service behavior while configuring the receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

Starting the Application

You must start the BizTalk application for polling Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleDB one-way receive port, which polls Oracle using the stored procedure specified for the **PollingStatement** binding property, is running.
- The FILE send port, which receives messages from Oracle database, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** which returns a positive value indicating the adapter to execute the statement specified for **PollingStatement** binding property.
- The adapter executes the GET_ACTIVITYS stored procedure specified for the **PollingStatement** binding property and returns all the rows in the ACCOUNTACTIVITY table. The response from Oracle database resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<GET_ACTIVITYS xmlns=" http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/PollingPackage/ACCOUNT_PKG">
  <OUTRECS>
    <OUTRECSRecord xmlns=" http://Microsoft.LobServices.OracleDB/2007/03/ReferencedRecordTypes/SCOTT/ACCOUNT_PKG/GET_ACTIVITYS/SCOTT/GET_ACTIVITYS">
      <TID>1</TID>
      <ACCOUNT>100001</ACCOUNT>
      <AMOUNT>500</AMOUNT>
      <DESCRIPTION />
      <TRANSDATE>2008-06-21T15:52:19</TRANSDATE>
      <PROCESSED>n</PROCESSED>
    </OUTRECSRecord>
    <OUTRECSRecord xmlns=" http://Microsoft.LobServices.OracleDB/2007/03/ReferencedRecordTypes/SCOTT/ACCOUNT_PKG/GET_ACTIVITYS/SCOTT/GET_ACTIVITYS">
      .....
      .....
    </OUTRECSRecord>
    .....
    .....
  </OUTRECS>
</GET_ACTIVITYS>
```

- The adapter executes the post-poll statement, which moves all the data from ACCOUNTACTIVITY table to another table.
- After the polling interval, the adapter again executes **PolledDataAvailableStatement**. Because ACCOUNTACTIVITY table has no records now, **PolledDataAvailableStatement** does not return a positive value and hence the adapter does

not execute the statement specified for the **PollingStatement** binding property. As a result, adapter client does not get any polling message.

- The adapter client will not get any more polling messages until some records are explicitly inserted into the ACCOUNTACTIVITY table. To insert more records, you can run the more_activity_data.sql script provided with the samples. After you run this script, the next time **PolledDataAvailableStatement** is executed, it returns a positive value. As a result, the adapter executes the polling statement and adapter clients again receive a polling message.

Note

The Oracle Database adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling Oracle Database by Using BizTalk Server](#)

Performing Composite Operations on Oracle Database by Using BizTalk Server

The Oracle Database adapter enables adapter clients to perform composite operations on Oracle database. A composite operation can include:

- Insert, Update, Delete, and Select operations on tables and views.
- Stored procedures and functions, inside or outside a package.

A single composite operation can have any number of these operations, in any order. For example, you can have two inserts followed by a delete, and finally a stored procedure execution. Also, you can have different operations targeting different database tables or views. For more information about how the adapter supports composite operations, see [Performing Composite Operations](#). For information about the structure of the SOAP message for composite operations, see [Message Schemas for the Composite Operation](#).

How to Perform Composite Operations on Oracle Database?

Performing an operation on Oracle database using Oracle Database adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform composite operations on Oracle database, these tasks are:

1. Create a BizTalk project in Visual Studio and generate schema for all the operations you want to invoke.
2. Manually create a schema file that includes references to all the schemas you generated in the previous step.
3. Create messages in the BizTalk project for sending and receiving messages from Oracle database. These messages must conform to the request and response schema you created in the previous step.
4. Create an orchestration to invoke the composite operation on Oracle database.
5. Build and deploy the BizTalk project.
6. Configure the BizTalk application by creating physical send and receive ports.
7. Start the BizTalk application.

This topic provides instructions on how to perform these tasks.

Generating Schema

In this topic, to demonstrate how to perform composite operations, we will perform the following tasks in the same order:

- Insert record into the ACCOUNTACTIVITY table.
- Retrieve all the records in the ACCOUNTACTIVITY table by invoking the GET_ALL_ACTIVITY procedure within the ACCOUNT_PKG package.
- Delete the record from the ACCOUNTACTIVITY table.

Run the scripts provided with the samples to create the ACCOUNTACTIVITY table. For more information about the samples, see [Samples](#).

You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate schemas.

Creating a Composite Schema Definition

You must now create a composite schema in the Visual Studio BizTalk project that references the schemas you created for the individual operations. Perform the following steps to create a composite schema definition.

To add a composite schema definition

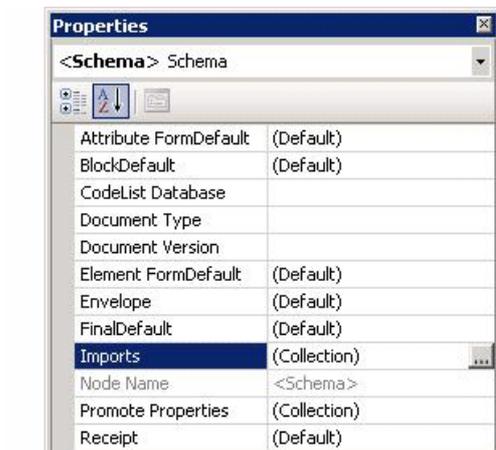
1. Add a schema file to the BizTalk project in Visual Studio. Right-click the solution name, point to **Add**, and then click **New Item**. In the **Add New Item** dialog box, from the **Categories** box, click **Schema Files**. From the **Templates** box, click **Schema**. Specify a name for the schema file and click **OK**.

For this example, specify the schema file name as **CompositeSchema.xsd**.

2. Add references to the schema generated for the different operations that you want to perform. In this example, the different schemas generated for operations are:
 - OracleDBBinding.xsd, for Insert and Delete operations on ACCOUNTACTIVITY table.
 - OracleDBBinding2.xsd, for the GET_ALL_ACTIVITY procedure.

To add references:

- a. Right-click the root **<Schema>** node in the CompositeSchema.xsd, and click **Properties**.
- b. In the **Property** box, click the ellipsis button (...) against the **Imports** property.



- c. In the **Imports** dialog box, from the **Import new schema as** list, select **XSD Import**, and then click **Add**.
- d. In the **BizTalk Type Picker** dialog box, expand the BizTalk project name node, expand **Schemas**, and then select the schema you want to import. For this example, select **<BizTalk_project_name>.OracleDBBinding.xsd**. Click **OK**.

Repeat this step to import **<BizTalk_project_name>.OracleDBBinding2.xsd** too.

- e. In the **Imports** dialog box, click **OK**.
3. Add two child nodes to the root schema node. One child node corresponds to the request schema for performing the composite operation. The other child node corresponds to the response schema. The node that corresponds to the request schema can have any name. The node that corresponds to the response schema must be called **<request_schema_node>Response**. For this example, we will call the request schema node as **Request**. So, the response schema node is called **RequestResponse**.

Note

By default, a **Root** node is also added to a new schema file. You can rename the **Root** node to **Request**. To rename a node, right-click the node name and click **Rename**.

To add a node under the **<Schema>** node:

- a. Right-click the **<Schema>** node, point to **Insert Schema Node**, and click **Child Record**.

b. Rename the new node to **RequestResponse**.

4. Add child nodes under the **Request** node that correspond to the request schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:

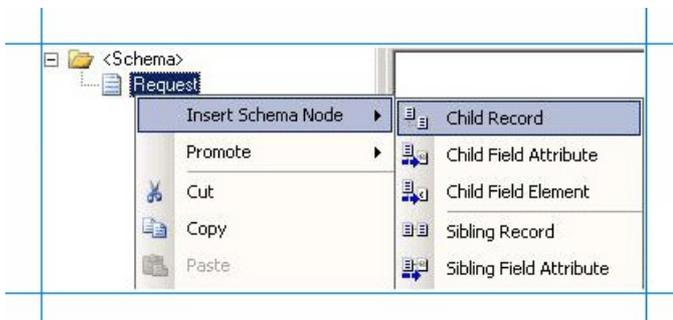
- Insert and Delete operations on the ACCOUNTACTIVITY table.
- GET_ALL_ACTIVITY procedure.

Important

You must add the nodes in the same order in which you want to perform the operations. For example, if you want to insert a record, then execute a stored procedure, and then delete a record you must first add a node for the Insert operation, followed by a node for the stored procedure, and finally a node for the Delete operation.

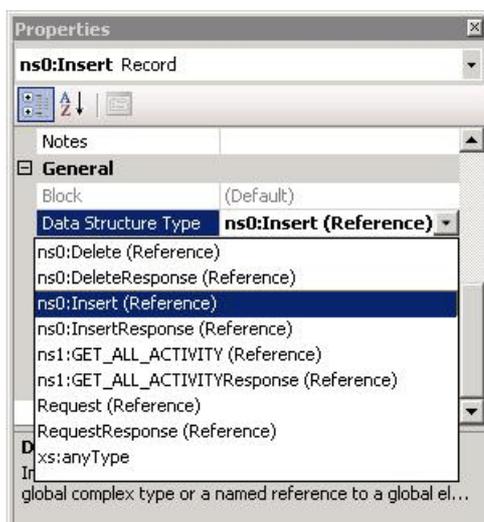
To add child nodes to the **Request** node:

a. Right-click the **Request** node, point to **Insert Schema Node**, and then click **Child Record**.



b. Rename the record to correspond to a request schema for an operation that you perform as part of the composite operation. For example, rename the node to "Insert".

c. Map the **Insert** node to the request schema for the Insert operation on the ACCOUNTACTIVITY table. To do so, right-click the **Insert** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **Insert (Reference)**.



d. Repeat these steps to add nodes for the request schemas for GET_ALL_ACTIVITY stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following table.

Node name	Mapped to schema
GET_ALL_ACTIVITY	GET_ALL_ACTIVITY (Reference)
Delete	Delete (Reference)

5. Add child nodes under the **RequestResponse** node that correspond to the response schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:

- Insert and Delete operations on the ACCOUNTACTIVITY table.
- GET_ALL_ACTIVITY stored procedure.

◆ Important
You must add the child nodes in the same order as the child nodes under the Request node.

To add child nodes to the **RequestResponse** node:

- Right-click the **RequestResponse** node, point to **Insert Schema Node**, and click **Child Record**.
- Rename the record to correspond to a response schema for an operation that you perform as part of the composite operation. For example, rename the node to "InsertResponse".
- Map the **InsertResponse** node to the response schema for the Insert operation on the ACCOUNTACTIVITY table. To do so, right-click the **InsertResponse** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **InsertResponse (Reference)**.
- Repeat these steps to add nodes for the response schemas for the GET_ALL_ACTIVITY stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following table.

Node name	Mapped to schema
GET_ALL_ACTIVITYResponse	GET_ALL_ACTIVITYResponse (Reference)
DeleteResponse	DeleteResponse (Reference)

6. Save the **CompositeSchema.xsd** file.

Defining Messages and Message Types

The composite schema that you created in the last step describes the "types" required for the messages in an orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schema you created in the previous step.

To create messages and link to schema

- Add an orchestration to the BizTalk project in Visual Studio. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
- Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
- In Orchestration View, right-click **Messages**, and then click **New Message**.
- Right-click the newly created message, and then select **Properties Window**.
- In the **Properties** pane for the **Message_1**, do the following:

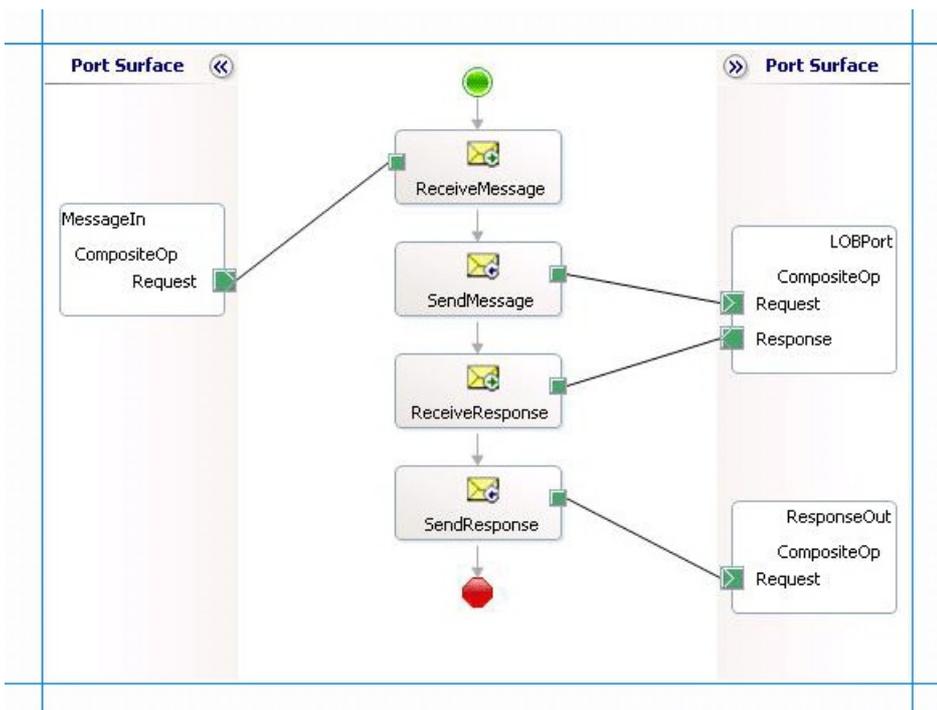
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>Composite_Op.CompositeSchema.Request</i> , where <i>Composite_Op</i> is the name of your BizTalk project. <i>CompositeSchema</i> is the schema you created manually for the composite operations.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>Composite_Op.CompositeSchema.RequestResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing composite operations on Oracle database. In this orchestration, you drop a request message at a defined receive location. The request message must conform to the composite schema you created earlier. The Oracle Database adapter consumes this message and passes it on to Oracle database. The response from Oracle database is saved to another location. You must include Send and Receive shapes to send messages to Oracle database and receive responses, respectively. A basic orchestration for performing composite operations resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>

ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.CompositeOp.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.CompositeOp.Request</i>

ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.CompositeOp.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.CompositeOp.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from Oracle database.
 - Define a physical WCF-Custom or WCF-OracleDB send port to send messages to Oracle database. Because the operations that are being as part of the composite operation are executed in a single transaction, make sure the **UseAmbientTransaction** binding property is set to **True**.

You must also specify the action in the send port. The action for a composite operation is "http://Microsoft.LobServices.OracleDB/2007/03/CompositeOperation". For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). For more information about how to specify actions for ports, see [Specifying SOAP Action](#).

<p> Note</p> <p>Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see Configuring a Physical Port Binding by Using a Port Binding File. If you import this binding file, the action on the send port is set to a dynamic action involving all the operations you selected in the Consume Adapter Service Add-in while generating the schema. For a composite operation, you must replace the dynamic action with "http://Microsoft.LobServices.OracleDB/2007/03/CompositeOperation".</p>
--

Starting the Application

You must start the BizTalk application for performing composite operations on Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.

- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleDB send port to send messages to Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the composite operations you created earlier. For example, a request message that inserts a record in the ACCOUNTACTIVITY table, invokes the GET_ALL_ACTIVITY stored procedure, and deletes a record from the ACCOUNTACTIVITY table is:

```
<Request xmlns="http://Composite_Op.CompositeSchema">
  <Insert xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY"
  >
    <RECORDSET>
      <ACCOUNTACTIVITYRECORDINSERT>
        <TID>1</TID>
        <ACCOUNT>100001</ACCOUNT>
        <AMOUNT>1500</AMOUNT>
        <DESCRIPTION></DESCRIPTION>
        <TRANSDATE>2008-06-21T15:52:19</TRANSDATE>
        <PROCESSED>n</PROCESSED>
      </ACCOUNTACTIVITYRECORDINSERT >
    </RECORDSET>
  </Insert>
  <GET_ALL_ACTIVITY xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCO
  UNT_PKG"/>
  <Delete xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY"
  >
    <FILTER>WHERE AMOUNT = 1500</FILTER>
  </Delete>
</Request>
```

The preceding request message first inserts a record and then invokes the GET_ALL_ACTIVITY procedure to get all the records in the ACCOUNTACTIVITY table. Then, the inserted record is deleted by specifying a FILTER clause. See [Message Schemas for the Composite Operation](#) for more information about the request message schema for performing composite operations on Oracle database using the Oracle Database adapter.

The orchestration consumes the message and sends it to Oracle database. The response from Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the preceding request message resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<RequestResponse xmlns="http://Composite_Op.CompositeSchema">
  <InsertResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOOT/Table/ACCOUNTA
  CTIVITY">
    <InsertResult>1</InsertResult>
  </InsertResponse>
  <GET_ALL_ACTIVITYResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Pack
  age/ACCOUNT_PKG">
    <ALLRECS>
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="
  urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                    <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />

```

```

        <xs:element minOccurs="0" name="AMOUNT" type="xs:decimal" />
        <xs:element minOccurs="0" name="DESCRIPTION" type="xs:string" />
        <xs:element minOccurs="0" name="TRANSDATE" type="xs:dateTime" />
        <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
    <NewDataSet xmlns="">
        <NewTable>
            .....
            .....
        </NewTable>
        .....
        .....
        <NewTable>
            <TID>10</TID>
            <ACCOUNT>100001</ACCOUNT>
            <AMOUNT>1000</AMOUNT>
            <TRANSDATE>2008-07-28T21:39:57</TRANSDATE>
            <PROCESSED>n</PROCESSED>
        </NewTable>
    </NewDataSet>
</diffgr:diffgram>
</ALLRECS>
</GET_ALL_ACTIVITYResponse>
<DeleteResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTA
CTIVITY">
    <DeleteResult>1</DeleteResult>
</DeleteResponse>
</RequestResponse>

```

The preceding response contains multiple result sets corresponding the different operations performed as part of the composite operation. For example, the `InsertResult` element contains '1', indicating the number of rows inserted by the Insert operation. Similarly, the `DeleteResult` element contains '1', indicating the number of rows deleted by the Delete operation.

◆ Important

If you experience time-out issues while executing a composite operation then it could be because the number of connections is less than the number of operations in a composite operation involving:

- Stored procedures containing BFILE, BLOB, CLOB, NCLOB, and REF CURSOR as OUT or IN OUT parameters.
- Select operation.

To resolve this issue, you must ensure that if there are "n" number of such operations in a composite operation, the value specified for the **MinPoolSize** binding property is "n+1" or greater. For more information about the **MinPoolSize** binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Receiving Database Change Notifications Using BizTalk Server

You can configure the Oracle Database adapter to receive database change notification messages from the Oracle database. You can specify a SELECT statement that the adapter uses to register for notifications with the Oracle database. The adapter receives a notification message when the result set for the SELECT statement, registered for notification, changes. For more information about how the adapter supports notification, see [Receiving Database Change Notifications](#).

Following are some scenarios in which you can configure the Oracle Database adapter with BizTalk Server to receive notifications from the Oracle database:

- Adapter clients get only “incremental” notification, for example, only for those changes that were made to a database table since the last notification.
- If large number of rows are inserted into a database table, the adapter clients can configure multiple receive locations to load-balance receiving notifications.

Once the adapter clients receive a notification message, they can perform specific tasks based on the kind of notification received. For example, a BizTalk orchestration can be designed in such a way that it performs one set of tasks if an insert notification is received and another set of tasks if an update notification is received.

Caution

If there is a network outage between the Oracle database and the adapter client, the notifications will not be sent to the adapter clients for the changes done on the Oracle database during the period of network outage, and thereafter. Therefore, you must use the Polling operation instead of the Notification operation for critical scenarios.

The topics in this section provide information on how to configure the adapter for each of these scenarios. To start getting notifications from the Oracle database using the Oracle Database adapter, you must specify certain binding properties. For more information about the binding properties related to notifications, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For more information about structure of notification messages, see [Message Schemas for the Notification Operation](#).

For receiving notifications from the Oracle database, make sure:

- You use the adapter to connect to Oracle database version 10.2 or later. Oracle database versions prior to 10.2 do not support notifications.
- The credentials you use to connect to Oracle for notifications has `change notification` privilege. This privilege is required for receiving database change notifications. To do so, connect to Oracle database using administrative privileges and then type the following command on the SQL prompt.

```
grant change notification to <user name>
```

- Decide on a TCP port you want ODP.NET to use for receiving database change notifications from Oracle database. Add that port to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkID=196959>. You must provide the same port number for the **NotificationPort** binding property. For more information about the binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

In This Section

- [Considerations for Receiving Database Change Notifications Using the Adapter](#)
- [Processing Notification Messages to Perform Specific Tasks](#)
- [Receiving Database Change Notifications Incrementally Using BizTalk Server](#)

- [Receiving Database Change Notifications On Multiple Receive Locations](#)
- [Receiving Database Change Notifications After a Receive Location Breakdown](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Considerations for Receiving Database Change Notifications Using the Adapter

This topic provides some considerations and best practices that you must keep in mind while using the Oracle Database adapter to receive database notifications from an Oracle database.

Considerations While Using the Adapter to Receive Notifications

You must consider the following while using the Oracle Database adapter to receive query notifications.

- The Oracle Database adapter simply passes on the notification, which it receives from the Oracle database, to the adapter clients. The adapter does not distinguish between the notifications for different operations, i.e., the adapter does not have any information whether a particular notification is for an Insert operation or an Update operation.
- The notification message for an operation is not affected by the number of records affected by that operation. For example, irrespective of the number of records inserted in an Oracle database table, the adapter clients receive only one notification message.
- We recommend that the adapter client application contain the logic to interpret the kind of notification received from the Oracle database. The adapter client applications can do so by extracting the information in the **<Info>** element of the received notification message. Here's an example of a notification message received for an Insert operation.

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>1</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Insert</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

Notice the value within the **<Info>** element. This value provides information on the operation for which the notification message was received. Your application should have the functionality to extract the value within the **<Info>** element and then based on the value, perform subsequent tasks. The topic [Processing Notification Messages to Perform Specific Tasks](#) has instructions on how to extract the value within the **<Info>** element.

- Ideally, after the client application receives a notification, it should update the record for which the notification is already received so that the subsequent notifications are not for the same record. For example, consider an **ACCOUNTACTIVITY** table that has a **Processed** column. For all new records inserted into the **ACCOUNTACTIVITY** table, the value in the **Processed** column is always 'n'. For example, after an insert operation, the records in the **ACCOUNTACTIVITY** table will look like the following:

Account Transaction ID	Processed
10001	n

To get notifications for the newly inserted record, the adapter client will set the **NotificationStatement** binding

property as:

```
SELECT * FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'
```

After, receiving the notification, the client application must set the value of the **Processed** column to 'y' so that the notification statement does not operate on the record that was already notified for. So, to achieve this, the client application must perform an Update operation on the **ACCOUNTACTIVITY** table. After the Update operation, the same record in the **ACCOUNTACTIVITY** table will look like the following:

Account Transaction ID	Processed
10001	y

Interestingly, the Update operation will again send a notification to the adapter client and the whole process will be repeated again. So, the client application must have the required logic to discard such unwanted notifications.

- If the **NotifyOnListenerStart** binding property is true, the adapter will send a notification to the adapter client every time the receive location starts. For more information on how to use the binding property and interpret the notification message, see [Receiving Database Change Notifications After a Receive Location Breakdown](#).

Typical Orchestration for Receiving Notifications

This section outlines the typical orchestration flow for receiving notifications using the Oracle Database adapter.

1. The first thing that the orchestration must do is to check the kind of notification received. The things to check for are:
 - Whether the notification was received for the receive location restart.
 - Whether the notification was received for an operation on a database table, such as Insert, Update, or Delete.

The orchestration must include an **Expression** shape, and within that an xpath query, to decide what kind of message is received.

2. After the notification type is available, the orchestration must include a decision block to perform specific actions based on the type of notification received. To achieve this, the orchestration must include a **Decide** shape. The **Decide** shape consists of a **Rule** block and an **Else** block. Within the **Rule** block, you must specify the condition and then include orchestration shapes to perform certain operations if the condition is met. Within the **Else** block, you must include orchestration shapes to perform certain operations if the condition is *not* met.

The preceding recommendations are described in detail in [Processing Notification Messages to Perform Specific Tasks](#).

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Processing Notification Messages to Perform Specific Tasks

You can use the Oracle Database adapter to receive notifications for changes to the Oracle database tables. However, the adapter only sends you a notification that some records were inserted, updated, or deleted in a certain database table. Any post-processing on those records must be handled by the client applications themselves. This topic presents a scenario-based description on how to process the records in the table based on the kind of notification received from the Oracle database.

Scenarios for Performing Subsequent Actions After Receiving Notification

Following are a couple of scenarios in which the adapter clients must perform certain post-notification tasks.

- **Scenario 1.** Consider a scenario where the adapter client must perform certain tasks based on the kind of notification you receive from the Oracle database. For example, the client application must update the records in table "A" if records are inserted in table "B". Similarly, the client application must delete records from table "A" if records are deleted from table "B".

In this scenario, from the notification message received, the adapter clients must extract the type of notification to decide whether the notification was for an insert operation or a delete operation. Once the notification type is ascertained, the adapter clients must perform subsequent actions to insert or update the relevant tables.

- **Scenario 2.** Consider a scenario where the receive location that receives notification messages for changes to a table goes down. While the receive location is down, some records are added to the table. However, for these records the adapter client does not receive any notification. When the receive location is back up, the adapter notifies the client by sending a specific message, and then the client application must look for all the records that were inserted in the database table while the receive location was down.

In this scenario, from the notification message received, the adapter clients must extract the information regarding whether the notification is for a change to a database table or for the receive location starting. If the notification is for the receive location starting, the adapter clients must implement the logic to process the records that might have been inserted, updated, or deleted while the receive location was down.

Note

These are just some example scenarios that are listed for a better understanding of how to use the notification feature in the Oracle Database adapter. However, the basic set of tasks required to extract the type of notification received will be similar for all scenarios. This topic provides instructions on how to extract the type of notification from a notification message.

How This Topic Demonstrates Receiving Notification Messages

In this topic, to demonstrate how to process notification messages to perform subsequent tasks, we consider a basic scenario where an adapter client uses BizTalk application to receive notification messages for changes to the ACCOUNTACTIVITY table. After the notification is received, the client filters the type of notification received and performs subsequent action. To demonstrate a very basic scenario, let us consider that the adapter client copies the notification messages to different folders based on the kind of notification received. Therefore:

- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.
- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotificaiton folder.

To achieve this as part of a BizTalk application, the orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the information about the kind of notification message received.

- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.

Configuring Notifications with the Oracle Database Binding Properties

The following table summarizes the Oracle Database adapter binding properties that you use to configure receiving notifications from the Oracle database. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

Note	
You may choose to specify these binding properties when generating the schema for the Notification operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-OracleDB receive port with the binding properties already set. For more information about creating a WCF-custom or WCF-OracleDB port using the binding file, see Configuring a Physical Port Binding by Using a Port Binding File .	

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the SELECT statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified SELECT statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For a complete description of how to use the Oracle Database adapter to receive notifications from the Oracle database, read further.

How to Receive Notification Messages from Oracle Database

Performing an operation on the Oracle database using Oracle Database adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** inbound operation. Optionally, you can specify values for the **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from the Oracle database.
3. Create an orchestration as described in the preceding section.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Note	
For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-OracleDB receive port. Two-way receive ports are not supported for inbound operations.	

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Notification** inbound operation. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive notifications from the Oracle database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>Process_Notification.OracleDBBinding.Notification</i> , where <i>Process_Notification</i> is the name of your BizTalk project. <i>OracleDBBinding</i> is the schema generated for the Notification operation.

Setting up the Orchestration

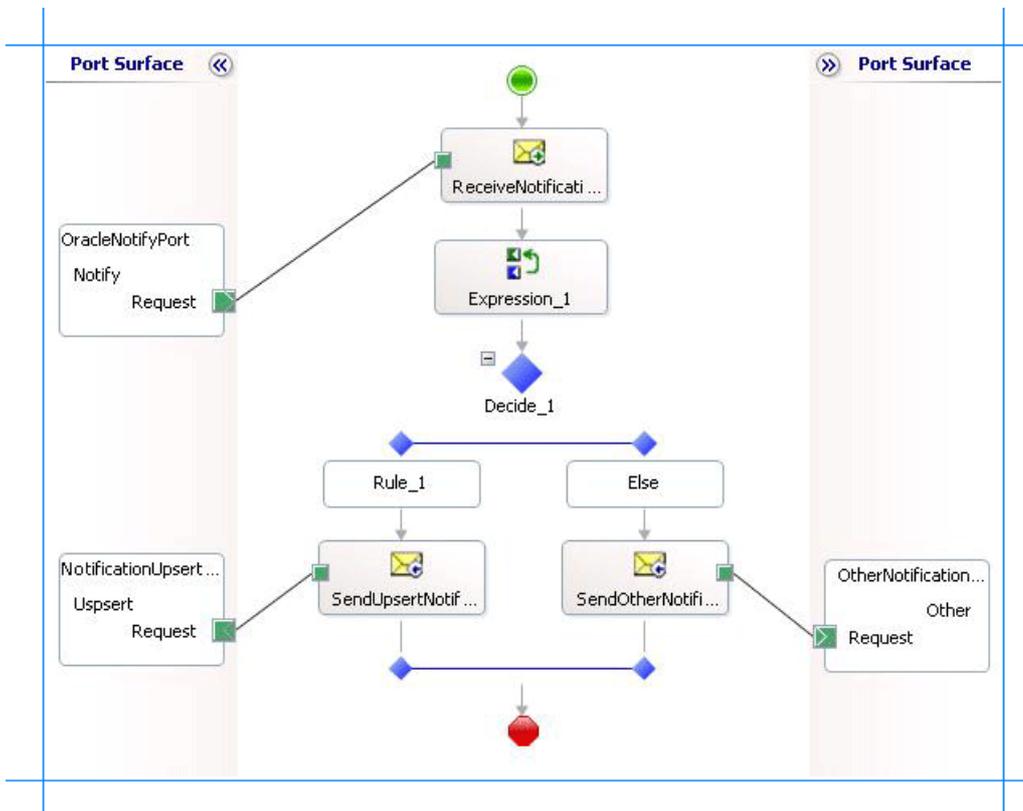
You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from the Oracle database and then performing tasks based on the type of notification received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The xpath query specified within the Expression shape extracts the type of notification into a variable, say **NotificationType**. The Decide shape uses the value in this variable to decide on the kind of notification received and takes the appropriate "path" to perform subsequent operations. As mentioned in the preceding section, the orchestration will perform the following operations based on the kind of notification message received.

- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.
- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotificaiton folder.

So, your orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the kind of notification received.
- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.
- Receive shape.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveNotification</i> • Set Activate to <i>True</i>

Adding an Expression Shape

The purpose of including an Expression shape in the orchestration is to have an xpath query to extract the kind of notification message received. Before creating an xpath query, let us look at the format of a notification message. A typical notification message resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>1</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Insert</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.ACCOUNT
ACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

As you see, the information about the type of the notification is available within the `<info>` tag, within the parent `<Notification>` tag. So, as part of this expression shape you must:

- Create a variable that contains the value within the `<Info>` tag and set its type to `System.String`. For more information about creating variables, see <http://go.microsoft.com/fwlink/?LinkId=102357>.

For this topic, name the variable as **NotificationType**.

- Create an xpath query to extract the value from the `<Info>` tag. The xpath query will resemble the following:

```
NotificationType = xpath(NotifyReceive, "string(/*[local-name()='Notification']/*[local
-name()='Info']/text())");
```

In this xpath query, **NotifyReceive** is the message you created for receiving notification messages. The excerpt within the `string` function indicates that the query must extract the value within the `<Info>` tag, which in turn is within the `<Notification>` tag. Finally, the value extracted by the query is assigned to the **NotificationType** variable.

Adding a Decide Shape

The purpose of adding a Decide shape is to include a decision block in the orchestration to decide what subsequent operations to perform based on the kind of notification message received. The decision is made on the basis of the value of the **NotificationType** variable. In this topic, the orchestration makes a decision based on the kind of notification message received. So, the condition in the Rule shape is specified as follows:

```
NotificationType.Equals("Insert") | NotificationType.Equals("Update")
```

This condition suggests that if the value for **NotificationType** variable is Insert or Update, the orchestration will perform one set of tasks. If the value of **NotificationType** variable is anything else, the orchestration will perform other set of tasks.

As mentioned in the preceding sections, to demonstrate a simple approach, the orchestration will copy messages to different folders based on the notification message type. So, within the Rule and Else blocks, you must add Send shapes to send the messages to different ports. For this topic, name the Send shape in the Rule block as **SendUpsertNotification** and the Send shape in the Else block as **SendOtherNotification**.

Adding Ports

You must now add the following logical ports to the orchestration:

- One-way receive port to receive notification messages from the Oracle database.
- One-way send port to send notification messages for Insert and Update operations to a specific folder.
- One-way send port to send notification messages for any other operations to a specific folder.

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleNotifyPort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleNotifyPort</i> • Set Type to <i>OracleNotifyPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
NotificationUpsertPort	<ul style="list-style-type: none"> • Set Identifier to <i>NotificationUpsertPort</i> • Set Type to <i>NotificationUpsertPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>
OtherNotificationPort	<ul style="list-style-type: none"> • Set Identifier to <i>OtherNotificationPort</i> • Set Type to <i>OtherNotificationPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>OracleNotifyPort.Notify.Request</i>
SendUpsertNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>NotificationUpsertPort.Upsert.Request</i>
SendOtherNotification	<ul style="list-style-type: none"> • Set Message to <i>Select</i> • Set Operation to <i>OtherNotificationPort.Other.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-OracleDB one-way receive port. This port listens for notifications coming from the Oracle database. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Make sure you specify the following binding properties for the receive port.

◆ Important
 You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-OracleDB receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>◆ Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre style="border: 1px solid black; padding: 5px; margin: 5px 0;">SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌 Note You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

📌 **Note**

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle Database adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-OracleDB receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the Oracle database for Insert and Update operations. Configure this port to drop notification messages to the folder C:\TestLocation\UpsertNotification.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the Oracle database for all other operations. Configure this port to drop notification messages to the folder C:\TestLocation\OtherNotification.

Starting the Application

You must start the BizTalk application for receiving notification messages from the Oracle database and for performing the subsequent Select and Update operations. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleDB one-way receive port, which receives the notification messages from the Oracle database is running.
- The two FILE send ports, which receive messages from Oracle database, are running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you start the BizTalk orchestration, the following set of actions take place:

- Because the **NotifyOnListenerStart** binding property is set to **True**, you receive the following message:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Info>ListenerStarted</Info>
  <Source>OracleDBBinding</Source>
  <Type>Startup</Type>
</Notification>
```

Note that the value in the `<Info>` tag is "ListnerStarted". Hence, this message is received in C:\TestLocation\OtherNotification folder.

- Insert a record in the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>1</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Insert</Info>
  <ResourceNames>
```

```

    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>

```

Note that the value in the `<Info>` tag is "Insert". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Update a record in the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>32</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Update</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>

```

Note that the value in the `<Info>` tag is "Update". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Delete a record from the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>16</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Delete</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>

```

Note that the value in the `<Info>` tag is "Delete". Hence, this message is received in C:\TestLocation\OtherNotification folder.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

Performing Complex Operations After Receiving Notification Messages

For simplicity and better understanding, the orchestration in this topic copies messages to different folders based on the notification type. However, in real-world scenarios you might want to perform more complex operations. You can perform similar procedures as provided in this topic and build on them to perform the operations you wish. For example, you can change the orchestration to insert records in another table if you get a notification message for an Insert operation on the ACCOUNTACTIVITY table. In such a case, you can make appropriate changes within the Decide shape.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Receiving Database Change Notifications Incrementally Using BizTalk Server

Important

For the sake of brevity, this topic only describes how to receive notifications incrementally. In business scenarios, the orchestration must ideally include the logic to extract the kind of notification message received and then perform any subsequent operations. In other words, the orchestration described in this topic must be built on top of the orchestration described in [Processing Notification Messages to Perform Specific Tasks](#).

This topic demonstrates how to configure the Oracle Database adapter to receive incremental query notification messages from Oracle. To demonstrate incremental notifications, we consider a table, ACCOUNTACTIVITY, with a "Processed" column. When a new record is inserted to this table, the value of the "Processed" column is set to 'n'. You can configure the adapter to receive incremental notifications by doing the following:

- Register for notifications using a SELECT statement that retrieves all records that have "Processed" column as 'n'. You can do so by specifying the SELECT statement for the **NotificationStatement** binding property.
- For rows which have been notified for, update the "Processed" column to 'y'.

This topic demonstrates how to create a BizTalk orchestration and configure a BizTalk application to achieve this.

Configuring Notifications with the Oracle Database Adapter Binding Properties

The following table summarizes the Oracle Database adapter binding properties that you use to configure receiving notifications from the Oracle database. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

Note

You may choose to specify these binding properties when generating the schema for the **Notification** operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-OracleDB receive port with the binding properties already set. For more information about creating a receive port using the binding file, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the SELECT statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified SELECT statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For a complete description of how to use the Oracle Database adapter to receive notifications from the Oracle database, read further.

How This Topic Demonstrates Receiving Notification Messages

In this topic, to demonstrate how the Oracle Database adapter supports receiving incremental database change notification messages from the Oracle database, we will configure the adapter to receive notifications for changes to the ACCOUNTACTIVITY table. Let us assume that the ACCOUNTACTIVITY table has columns "TID", "Account", and "Processed". Whenever a new record is added, the value of the "Processed" column is set to 'n'. So, to get incremental notifications you will have to do the following tasks as part of the BizTalk orchestration:

- Get notification for all records where "Processed" is 'n'. You can do this by specifying a SELECT statement as a notification statement.
- After the notification is received for a certain record, set "Processed" to 'y'. You can do this by executing a stored procedure, PROCESS_RECORDS, which updates the "Processed" column.

To demonstrate receiving incremental notifications, we do the following:

- Generate schema for the **Notification** (inbound operation), and **PROCESS_RECORDS** (outbound operation) on the ACCOUNTACTIVITY table.
- Create an orchestration that has the following:
 - A receive location to receive notification messages. You can configure for notification by specifying the SELECT statement as:

```
SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'
```

Note

You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.

- A send port to update the rows for which notification has already been sent. You will execute the PROCESS_RECORDS stored procedure on this port to set the value of "Processed" column to 'y' for the records for which notification is received.

Note that this operation must be executed after receiving the notification messages so that the processed rows are updated. To do away with the overhead of waiting to get the notification response and then manually dropping a request message to execute the PROCESS_RECORDS procedure, you will generate the request message for PROCESS_RECORDS procedure within the orchestration itself. You can do so by using the **Construct Message** shape within an orchestration.

How to Receive Notification Messages from the Oracle database

Performing an operation on the Oracle database using Oracle Database adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** (inbound operation) and **PROCESS_RECORDS** procedure (outbound operation) on the ACCOUNTACTIVITY table. Optionally, you can specify values for the **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from the Oracle database.
3. Create messages in the BizTalk project for executing the PROCESS_RECORDS stored procedure and receiving response messages.
4. Create an orchestration that does the following:
 - Receives notification message from the Oracle database.

- Creates a message to execute the `PROCESS_RECORDS` procedure.
- Sends this message to the Oracle database to select and update the records and receive a response.

5. Build and deploy the BizTalk project.

6. Configure the BizTalk application by creating physical send and receive ports.

Note

For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-OracleDB receive port. Two-way receive ports are not supported for inbound operations.

7. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Notification** operation and **PROCESS_RECORDS** procedure. See [Retrieving Metadata for Oracle Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.
4. Select the contract type as **Client (Outbound operations)**.
5. Generate schema for the **PROCESS_RECORDS** procedure. This procedure is available under the **ACCOUNT_PKG** package.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create three messages—one to receive notifications from the Oracle database, one to execute the `PROCESS_RECORDS` procedure, and one to receive the response for the procedure.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>OracleNotifyIncremental.OracleDBBinding.Notification</i> , where <i>OracleNotifyIncremental</i> is the name of your BizTalk project. <i>OracleDBBinding</i> is the schema generated for the Notification operation.

6. Repeat step 3 to create two new messages. In the **Properties** pane for the new message, do the following:

Set Identifier to	Set Message Type to
Procedure	<i>OracleNotifyIncremental.OracleDBBinding1.PROCESS_RECORDS</i> , where <i>OracleDBBinding1</i> is the schema generated for the PROCESS_RECORDS procedure.
ProcedureResponse	<i>OracleNotifyIncremental.OracleDBBinding1.PROCESS_RECORDSResponse</i>

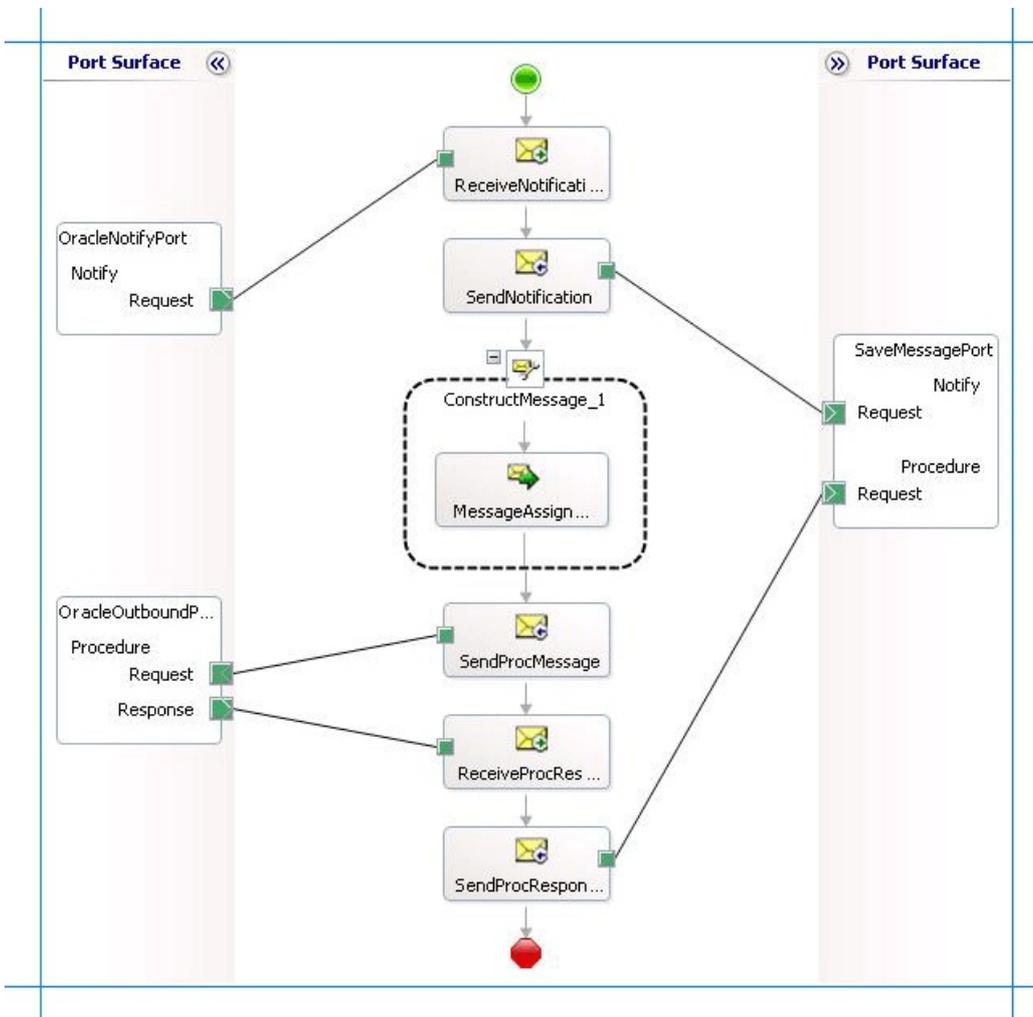
Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from the Oracle database and then updating the rows for which notification was received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The notification message is received at a FILE location. Once the response is received, the orchestration constructs a message to invoke the PROCESS_RECORDS procedure, which updates the rows for which notification is received. The response for this message is also received at the same FILE location.

So, your orchestration must contain the following:

- A one-way WCF-Custom or WCF-OracleDB receive port to receive notification messages.
- A two-way WCF-Custom or WCF-OracleDB send port to send messages to execute the PROCESS_RECORDS procedure.
- A **Construct Message** shape to construct messages, to execute PROCESS_RECORDS procedure, within the orchestration.
- A FILE send port to save the notification message and the response for the PROCESS_RECORDS procedure.
- Receive and send shapes.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveNotification</i> Set Activate to <i>True</i>
SaveNotification	Send	<ul style="list-style-type: none"> Set Name to <i>SaveNotification</i>
SendProcMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendProcMessage</i>
ReceiveProcResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveProcResponse</i>
SaveProcResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SaveProcResponse</i>

Adding Construct Message Shape

You can use the **Construct Message** shape to generate a request message within the orchestration to execute the PROCESS_RECORDS procedure. To do so, you must add a **Construct Message** shape and within that a **Message Assignment** shape to your orchestration. For this example, the **Message Assignment** shape invokes code that generates a message that is sent to the Oracle database to execute the procedure. The **Message Assignment** shape also sets the action for the message to be sent to the Oracle database.

For the construct message shape, set the **Message Constructed** property to **Procedure**.

The code to generate the response could be part of the same Visual Studio solution as your BizTalk project. A sample code for

generating a response message looks like this.

```
namespace SampleMessageCreator
{
    public class SampleMessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;

        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\MessageIn";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOption.To
                pDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }
            //Create Message From XML
            Message = new XmlDocument();
            Message.PreserveWhitespace = true;
            Message.Load(ResponseDoc);
            return Message;
        }
    }
}
```

For the above code excerpt to be able to generate a request message, you must have an XML request message (for the PROCESS_RECORDS procedure) in the location specified for the `XmlFileLocation` variable.

Note

After you build the project, `MessageCreator.dll` will be created in the project directory. You must add this DLL to the global assembly cache (GAC). Also, you must add the `MessageCreator.dll` as a reference in the BizTalk project.

Add the following expression to invoke this code from the **Message Assignment** shape and to set the action for message. To add an expression, double-click the **Message Assignment** shape to open the Expression Editor.

```
Procedure = SampleMessageCreator.SampleMessageCreator.XMLMessageCreator();
Procedure(WCF.Action) = "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/PROCESS_RECORDS";
```

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleNotifyPort	<ul style="list-style-type: none">• Set Identifier to <i>OracleNotifyPort</i>• Set Type to <i>OracleNotifyPortType</i>• Set Communication Pattern to <i>One-Way</i>• Set Communication Direction to <i>Receive</i>

SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i> • Create an operation <i>Notify</i>. This operation is used for notification messages. • Create an operation <i>Procedure</i>. This operation is used for select response messages.
OracleOutboundPort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleOutboundPort</i> • Set Type to <i>OracleOutboundPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>OracleNotifyPort.Notify.Request</i>
SaveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>SaveMessagePort.Notify.Request</i>
SendProcMessage	<ul style="list-style-type: none"> • Set Message to <i>Procedure</i> • Set Operation to <i>OracleOutboundPort.Procedure.Request</i>
ReceiveProcResponse	<ul style="list-style-type: none"> • Set Message to <i>ProcedureResponse</i> • Set Operation to <i>OracleOutboundPort.Procedure.Response</i>
SaveProcResponse	<ul style="list-style-type: none"> • Set Message to <i>ProedureResponse</i> • Set Operation to <i>SaveMessagePort.Procedure.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-OracleDB one-way receive port. This port listens for notifications coming from the Oracle database. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle Database Adapter](#). Make sure you specify the following binding properties for the receive port.

◆ Important

You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>◆ Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre>SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌 Note You must specify the table name along with the schema name. For example, <code>SCOTT.ACCOUNTACTIVITY</code>.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle Database adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-OracleDB receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a physical WCF-Custom or WCF-OracleDB send port to send messages to the Oracle database to execute the PROCESS_REOCRDS procedure. You must also specify the action in the send port.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the messages from the Oracle database. These will be the notification messages received from the Oracle database and messages for the PROCESS_RECORDS procedure you execute through the WCF-Custom or WCF-OracleDB send port.

Starting the Application

You must start the BizTalk application for receiving notification messages from the Oracle database and for executing the PROCESS_RECORDS procedure. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleDB one-way receive port, which receives the notification messages from the Oracle database is running.
- The WCF-Custom or WCF-OracleDB send port to execute the PROCESS_RECORDS procedure is running.
- The FILE send port, which receives messages from the Oracle database, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

Assume that the ACCOUNTACTIVITY table already has some records. Also, make sure the XML message to execute PROCESS_RECORDS procedure is available at C:\TestLocation\Messageln. The XML file should resemble the following:

```
<PROCESS_RECORDS xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG"/>
```

Once the BizTalk orchestration is started, the following set of actions take place, in the same sequence:

- The adapter receives a notification message that resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Info>ListenerStarted</Info>
  <Source>OracleDBBinding</Source>
  <Type>Startup</Type>
</Notification>
```

This message notifies that the receive port for receiving the notification messages is started. Note that the value for the <Info> element is "ListnerStarted".

- The adapter executes the PROCESS_RECORDS procedure. The next response from the Oracle database is for the procedure.

```

<?xml version="1.0" encoding="utf-8" ?>
<PROCESS_RECORDSResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Pa
ckage/ACCOUNT_PKG">
  <TABLE_DATA>
    <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdat
a="urn:schemas-microsoft-com:xml-msdata">
      <xs:element msdata:IsDataSet="true" name="NewDataSet">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                  <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
      <NewDataSet xmlns="">
        <NewTable>
          <TID>1</TID>
          <ACCOUNT>100001</ACCOUNT>
          <PROCESSED>n</PROCESSED>
        </NewTable>
        <NewTable>
          .....
          .....
        </NewTable>
        .....
        .....
      </NewDataSet>
    </diffgr:diffgram>
  </TABLE_DATA>
</PROCESS_RECORDSResponse>

```

This is the response for the SELECT statement execute as part of the PROCESS_RECORDS procedure.

- The PROCESS_RECORDS procedure also updates the rows to set PROCESSED to 'y'. Hence, the adapter receives another notification for the Update operation.

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>32</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
</Notification>

```

```

    </NotificationDetails>
  </Details>
  <Info>Update</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>

```

Note that the `Info` element contains "Update".

- After the second notification, the adapter again executes the `PROCESS_RECORDS` procedure. However, now because there are no records where `PROCESSED` column is set to 'n', the procedure returns an empty response resembling the following.

```

<?xml version="1.0" encoding="utf-8" ?>
<PROCESS_RECORDSResponse xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Pa
ckage/ACCOUNT_PKG">
  <TABLE_DATA>
    <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdat
a="urn:schemas-microsoft-com:xml-msdata">
      <xs:element msdata:IsDataSet="true" name="NewDataSet">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                  <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
      <NewDataSet xmlns="" />
    </diffgr:diffgram>
  </TABLE_DATA>
</PROCESS_RECORDSResponse>

```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Receiving Database Change Notifications On Multiple Receive Locations

Consider a scenario where you have multiple receive locations created as part of different BizTalk applications configured to receive query notifications for the same table (e.g. ACCOUNTACTIVITY) in the same database. If a hundred records are inserted into the same table, all the receive locations will get the notification message. To effectively receive notifications across multiple receive locations, you can call operations from your BizTalk application in such a way that if a notification is received by one receive location, the other receive location does not get the same notification. So, you can effectively load-balance notifications received on multiple locations.

The tasks required to set up an orchestration to load-balance receiving notifications are same as that for [Receiving Database Change Notifications Incrementally Using BizTalk Server](#). This topic lists the only the difference between the two approaches.

Load-Balancing Query Notifications Across Multiple Receive Locations

Like in the topic [Receiving Database Change Notifications Incrementally Using BizTalk Server](#), you configured incremental notifications by executing a PROCESS_RECORDS procedure. To configure load-balancing, you could execute a stored procedure that deletes the records that have been notified for. For example, consider a stored procedure NOTIFY_LOAD_BALANCE with the following definition:

```
PROCEDURE NOTIFY_LOAD_BALANCE (TABLE_DATA OUT SYS_REFCURSOR) IS
  var int;
BEGIN
  SELECT TID INTO var FROM ACCOUNTACTIVITY WHERE ROWNUM = 1 FOR UPDATE;
  OPEN TABLE_DATA FOR SELECT * FROM ACCOUNTACTIVITY WHERE TID = var;
  DELETE FROM ACCOUNTACTIVITY WHERE TID = var;
END NOTIFY_LOAD_BALANCE;
```

When you execute this stored procedure as part of the BizTalk application, the record for which notification is already received gets deleted. So, the other receive location gets notification for the next record.

Here are the high-level steps you must perform to configure load-balancing for receiving notifications.

1. Create schema for **Notification** (inbound operation) and **NOTIFY_LOAD_BALANCE** procedure (outbound operation).
2. Add an orchestration and add three messages for receiving notification, executing the procedure, and getting response for the procedure.
3. Create an orchestration by adding Send and Receive shapes, Construct Message shape, and ports. You can use the same sample code for constructing a message to invoke the NOTIFY_LOAD_BALANCE stored procedure. Note that while performing the operation in BizTalk Server Administration console, you must have the request message for the NOTIFY_LOAD_BALANCE procedure in the location C:\TestLocation\MessageIn. You do so because the code snippet you invoke as part of the orchestration created in [Receiving Database Change Notifications Incrementally Using BizTalk Server](#) creates a request message based on the request XML present in C:\TestLocation\MessageIn.
4. Build and deploy the application. To demonstrate load-balancing, you must deploy this orchestration at least on two different computers that have BizTalk Server and Oracle Database adapter installed.
5. In the BizTalk Server Administration console on both the computers, specify the following binding properties for the WCF-Custom or WCF-OracleDB receive location:

Binding Property	Value

InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p> Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre>SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p> Note You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p>
NotifyOnListenerStart	Set this to True .

6. Start the BizTalk application.
7. To start receiving notifications, insert a hundred records into the ACCOUNTACTIVITY table. While doing so, make sure the request XML for invoking the NOTIFY_LOAD_BALANCE procedure is available in C:\TestLocation\MessageIn.
8. Monitor the location (on both the computers) where the BizTalk application will be dropping the notification messages. You will notice that of the hundred records inserted, one location gets notifications for some records while the other location gets notification for the remaining records. Together, both the locations will get notification for all the hundred records.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Receiving Database Change Notifications After a Receive Location Breakdown

Consider a scenario where you have a BizTalk application that receives database change notification messages when changes are made to the ACCOUNTACTIVITY table. If the receive location configured as part of the BizTalk application breaks down, and simultaneously records are added into the ACCOUNTACTIVITY table, you will not receive notifications for the recently added records. You will also not know when the receive location is available again. The Oracle Database adapter exposes a binding property, **NotifyOnListenerStart**, that you can configure to get a notification that the receive location has recovered. You can specify the following values for the **NotifyOnListenerStart** binding property:

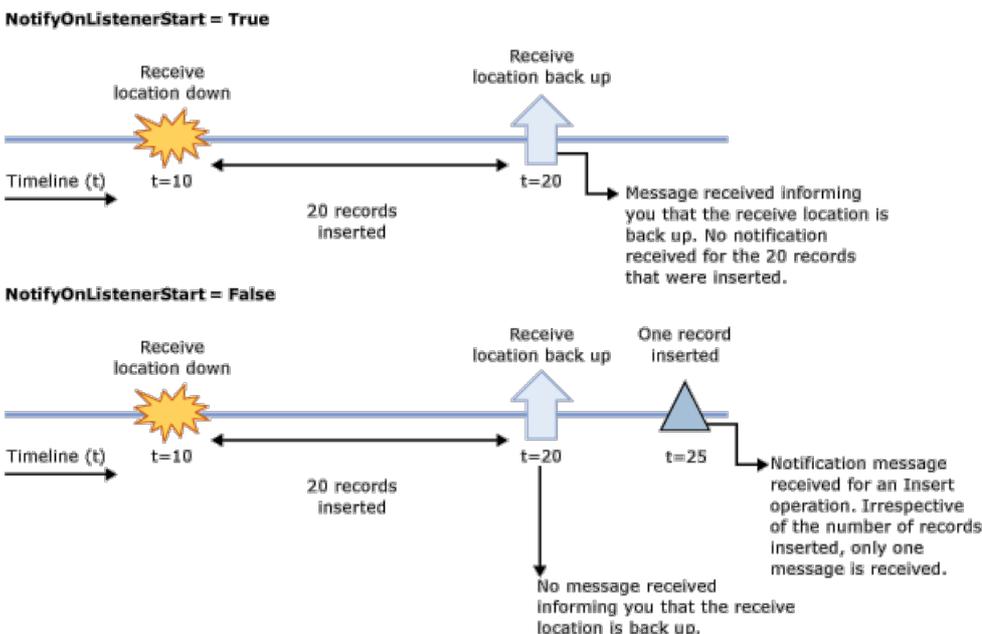
- Set this property to **True**, to receive a notification informing that the receive location is available, as soon as the receive location recovers.
- Set this property to **False**, to not receive a notification informing that the receive location has recovered, after the receive location recovers.

Default is **True**.

Configuring the Oracle Database Adapter Behavior

For either of the approaches, you do not need to perform any specific tasks while generating metadata or while configuring the BizTalk application. You only need to set the **NotifyOnListenerStart** binding property accordingly on the WCF-Custom or WCF-OracleDB receive location. To create the BizTalk application, you must perform the same set of tasks as described in [Receiving Database Change Notifications Incrementally Using BizTalk Server](#). However, when configuring the BizTalk application using BizTalk Server, you can try changing the value of **NotifyOnListenerStart** binding property and see the difference in the two configurations.

The following figure demonstrates how the notifications are received based on the value of the **NotifyOnListenerStart** binding property.



Note that in the first scenario, when the **NotifyOnListenerStart** is set to **True** and records are inserted into the database table while the receive location was down, the adapter only sends you a notification message when the receive location comes up. The adapter does not perform any operation to process the records that were inserted while the receive location was down. The adapter client must implement the relevant logic in their application to process the records that were inserted while the receive location was down.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Developing Applications by Using the WCF Service Model

At the lowest level, the Windows Communication Foundation (WCF) presents a programming model in which clients invoke operations on a service by exchanging SOAP messages over a channel established between client and service endpoints. This model, known as the WCF channel model, exposes data types and methods that enable you to operate directly on the WCF channel architecture. The WCF channel model provides you with direct control over the contents of the SOAP messages you create and over the way both your application and the Oracle Database adapter consume them; however, creating well-formed SOAP messages to send over a channel and validating the reply messages returned can be a detailed and exacting task.

For this reason, WCF provides another programming model called the WCF service model. The WCF service model involves the use of proxy classes to invoke operations on a target service or to receive operations from a client.

- The proxy class used to invoke operations on a target service is called a WCF client class. This class models the operations exposed by a service as .NET methods with strongly-typed parameters. By using the WCF service model, you can invoke the operations exposed by the Oracle Database adapter as .NET methods on the WCF client. For more information about WCF clients, see "WCF Client Overview" at <http://go.microsoft.com/fwlink/?LinkId=91458>.
- In the WCF service model, the service contract exposed by a service is represented by an interface. This managed code representation of the service contract is called a WCF service contract. The WCF service contract models operations as methods with strongly-typed parameters. To receive an operation from a client you implement a class, the WCF service, from this interface. You can then host an instance of this class in a **System.ServiceModel.ServiceHost** to enable a client to invoke the operation on your code. By using the WCF service model and a WCF service contract targeted to the POLLINGSTMT operation, you can receive the results of a polling operation on the Oracle database using Oracle Database adapter.

You use tools to generate a WCF client class or a WCF service contract and associated helper code from the service metadata that the Oracle Database adapter exposes. You can use either of the following tools:

- The ServiceModel Metadata Utility Tool (svcutil.exe), which ships with WCF
- The Add Adapter Service Reference Visual Studio Plug-in, which ships with the Oracle Database adapter

The Add Adapter Service Reference Plug-in is integrated with the Visual Studio design experience and presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations exposed by the adapter. For more information about how to generate a WCF client or a WCF service contract, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

Because it presents a model that is familiar to .NET programmers and that hides the underlying complexities of SOAP message exchange over a channel, the WCF service model is often the best choice to develop programming solutions for the Oracle Database adapter. However, there are scenarios in which the WCF channel model might be a better choice. For example, the WCF service model only supports streaming for the ReadLOB operation. This is because serializing and de-serializing between the XML representation of objects in a SOAP message and the .NET types used to represent them in the service model involves reading the entire message into memory. (The result of a ReadLOB operation is an exception to this rule.)

The WCF channel model provides support for XML node-level streaming on all operations and data-level streaming on the ReadLOB and UpdateLOB operations. If you are dealing with queries that return large result sets or are attempting to update an LOB field in a table, the WCF channel model might be a better choice. For more information about using the WCF channel model, see [Developing Applications by Using the WCF Channel Model](#).

The topics in this section contain information, procedures, and examples to help you create and use the WCF service model to develop applications by using the Oracle Database adapter.

In This Section

- [Overview of Using the WCF Service Model with the Adapter](#)
- [Metadata and the WCF Service Model](#)
- [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#)

- [Specifying a Client Binding for the Oracle Database](#)
- [Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model](#)
- [Performing Operations on Tables with Large Object Types by Using the WCF Service Model](#)
- [Invoking Functions and Procedures by Using the WCF Service Model](#)
- [Performing Operations Using REF CURSORS in the WCF Service Model](#)
- [Performing Operations Using RECORD Types in the WCF Service Model](#)
- [Performing a SQLEXECUTE Operation by Using the WCF Service Model](#)
- [Receiving Polling-based Data-changed Messages by Using the WCF Service Model](#)
- [Receiving Database Change Notifications Using the WCF Service Model](#)

Overview of Using the WCF Service Model with the Adapter

When you consume operations that the Microsoft BizTalk Adapter for Oracle Database surfaces, your code acts either as a client or a service to the adapter. For almost all of the operations that the Oracle Database adapter surfaces, your code is the client. That is, your application invokes the operation on the adapter; for example to insert records into an Oracle table. The only operation for which your code acts as a service to the Oracle Database adapter is for the POLLINGSMT operation. In this case, the adapter sends the results of the polling query operation to your application.

In the Windows Communication Foundation (WCF) service model, the service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The Oracle Database adapter and WCF provide tools that enable you to generate this interface for targeted operations from the metadata that the adapter exposes. These tools also create a WCF client class that can be used to invoke the operations exposed in the service interface. A client application can call the methods of the WCF client class to invoke operations on the adapter. To implement a service to receive the POLLINGSTMT operation from the Oracle Database adapter, you implement the interface generated for the POLLINGSTMT operation.

The following sections explain how to use the WCF service model to create client and service code for the Oracle Database adapter.

Creating and Invoking Operations on a WCF Client by Using the Oracle Database Adapter

To use the WCF service model to invoke operations on the Oracle Database adapter, you must first generate a WCF client class for the target operations. You can then create an instance of this class, a WCF client, and call its methods to perform operations on the Oracle database.

To invoke operations on the Oracle Database adapter

1. Generate a WCF client class and helper code. Use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class targeted at the Oracle database artifacts with which you want to work. For more information about how to generate a WCF client, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).
2. Create a WCF client instance by specifying a client binding. Specifying a client binding involves specifying the binding and endpoint address that the WCF client will use. You can do this either imperatively in code or declaratively in configuration. For more information about how to specify a client binding, see [Specifying a Client Binding for the Oracle Database](#). The following code creates a WCF client that can be used to perform data manipulation language (DML) operations on an Oracle database table (/SCOTT/ACCOUNTACTIVITY). It also sets the credentials for the Oracle database. The WCF client is initialized from configuration.

```
SCOTTTableACCOUNTACTIVITYClient aaTableClient =
    new SCOTTTableACCOUNTACTIVITYClient("OracleDBBinding_SCOTT.Table.ACCOUNTACTIVITY")
;

aaTableClient.ClientCredentials.UserName.UserName = "SCOTT";
aaTableClient.ClientCredentials.UserName.Password = "TIGER";
```

3. Open the WCF client.

```
aaTableClient.Open();
```

4. Invoke methods on the WCF client created in step 2 to perform operations on the Oracle database. The following code invokes the **Select** method of the WCF client to perform the following SQL SELECT query on the ACCOUNTACTIVITY table: `SELECT * FROM ACCOUNTACTIVITY`.

```
// create a record set parameter to hold the SELECT query result set and invoke the Se
lect operation;
microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRE
CORDSELECT[] selectRecords;
selectRecords = aaTableClient.Select("*", null);
```

5. Close the WCF client.

```
aaTableClient.Close();
```

For more information about performing DML operations on tables and views, including the Select operation used above, see [Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model](#).

Creating and Implementing a WCF Service by Using the Oracle Database Adapter

The Oracle Database adapter can perform polling on an Oracle database table or view. This functionality lets you specify a SQL SELECT query that the adapter should execute periodically against the Oracle database. The results of this query are returned to your application through a special operation, the POLLINGSTMT operation. To receive the results of the polling query, your application must implement the service contract that the Oracle Database adapter exposes for the POLLINGSTMT operation.

To implement a service to receive the POLLINGSTMT operation, you must first generate the .NET interface (also called the WCF service contract) that represents the service contract exposed by the Oracle Database adapter for the POLLINGSTMT operation. For more information about how to do this, see

[Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

Then you implement a WCF service by implementing the generated interface. This class contains the business logic to process the POLLINGSTMT message and return a response to the adapter. Then you use a service host

(**System.ServiceModel.ServiceHost**) to host an instance of this service. For more detailed information, see [Receiving Polling-based Data-changed Messages by Using the WCF Service Model](#).

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Metadata and the WCF Service Model

In the WCF service model, you use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutile.exe) to generate proxy classes through which your code can either:

- Invoke operations on the adapter (a WCF client class)
- Receive operations from the adapter (a WCF service contract)

These tools generate .NET classes that represent a service contract for target operations (as well as the supporting message contracts, operation contracts, and data contracts) from the metadata exposed by the Oracle Database adapter. For help in understanding the structure of this generated code, see "Understanding Generated Client Code" at <http://go.microsoft.com/fwlink/?LinkId=98365>. This topic specifically describes code that svcutil.exe generates, but its content is also applicable to the code that the Add Adapter Service Reference Plug-in generates. For information about how to generate a WCF client class or WCF service contract for target operations and about the differences between svcutil.exe and the Add Adapter Service Reference Plug-in, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts

You can use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class or a WCF service contract (interface) targeted at selected operations on Oracle database artifacts. You can also use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate the WCF client class or WCF service contract; however, the Add Adapter Service Reference Plug-in exposes the functionality of the ServiceModel Metadata Utility Tool through a standard Microsoft Windows interface. It also provides browse and search capabilities that are not available with the svcutil.exe tool, and it generates a configuration file based on the binding properties that you select when you connect to the Oracle database.

Generating a Client Class by Using the Add Adapter Service Reference Plug-in

Perform the following steps to generate a WCF client class by using the Add Adapter Service Reference Plug-in.

To generate a WCF client class

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for Oracle Operations in Visual Studio](#) to connect to the Oracle database and browse and search for operations. To create a WCF client class for the operations that you select, be sure that **Client (Outbound operations)** is selected from the **Select contract type** drop-down list (this is the default).
3. After you select all of the operations that you want to target, click **OK** to generate the WCF client class.

The Add Adapter Service Reference Plug-in adds two files to your project:

- **OracleDBBindingClient.cs**. This file contains the generated WCF client class and helper code for the operations that you selected.
- **App.config**. This file contains a binding configuration and client endpoint configurations. These configurations are based on the selections you made when you configured the binding and connection for the Add Adapter Service Reference Plug-in.

◆ Important

While using the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Generating a WCF Service Contract by Using the Add Adapter Service Reference Plug-in

The adapter exposes inbound operations to enable Oracle database to send messages to an adapter client. For such operations you must generate a WCF service contract. For example, the adapter exposes an inbound POLLINGSTMT operation to poll the Oracle database. The Oracle Database adapter executes the query specified by the **PollingStatement** binding property and sends the result set to the consuming application in a POLLINGSTMT message. In this scenario, the consuming application acts as a service and the Oracle Database adapter acts as the client. You must, therefore, implement a WCF service that can receive the POLLINGSTMT operation from the adapter. To do this, you use the Add Adapter Service Reference Plug-in to generate a .NET interface that represents the service contract that is surfaced by the adapter for the POLLINGSTMT operation. This .NET interface is also called a WCF service contract. You then implement this interface to create the WCF service that you can use to receive the POLLINGSTMT operation.

This section provides information on how to generate a WCF service contract using the Add Adapter Service Reference Plug-in for inbound operations exposed by the adapter.

To generate a WCF service contract for inbound operations

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for Oracle Operations in Visual Studio](#) to connect to the Oracle database. There are several binding properties and a URI property that you may want to set when you connect to the Oracle database for inbound

operations. For example, for the inbound polling operation (**POLLINGSTMT**), you must specify the **PollingStatement** binding property when you configure the connection to the Oracle database. The Oracle Database adapter uses the SQL SELECT statement specified in this property to generate the class that represents the result set returned by the POLLINGSTMT operation.

3. After you have connected to the Oracle database, select **Service (Inbound operations)** from the **Select contract type** drop-down list.
4. In the **Select a category** box, click the root node (*/*), and browse to the operation for which you want to generate the service contract. For example, for the polling operation, select **POLLINGSTMT** from the **Available categories and operations** box, and then click **Add**.
5. To generate the WCF service contract for the POLLINGSTMT operation, click **OK**.

The Add Adapter Service Reference Plug-in adds three files to your project:

- **OracleDBBindingInterface.cs.** This file contains the generated WCF service contract (interface) and helper code for the POLLINGSTMT operation.
- **OracleDBBindingService.cs.** This file contains a class that implements the interface defined in OracleDBBindingInterface.cs. You can implement the business logic that processes the records returned by the polling query in the POLLINGSTMT method in this class.
- **App.config.** This file contains a binding configuration, endpoint behaviors, and service endpoint configuration that are based on the selections you made when you configured the binding and connection for the Add Adapter Service Reference Plug-in.

◆ Important

While using the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Using svcutil.exe to Generate a WCF Client Class or a WCF Service Contract

You can use svcutil.exe to generate a WCF client class or a WCF service interface for your application. You must configure svcutil.exe to use it with the Oracle Database adapter. For more information about configuring and using svcutil.exe with the Oracle Database adapter, see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Oracle Database](#).

Svcutil.exe generates the WCF client class or WCF service contract in an output file. The default file name is output.cs. You must manually include this file in your Visual Studio project.

See Also

Concepts

[Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model](#)

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Specifying a Client Binding for the Oracle Database

After you have generated the WCF client class, you can create a WCF client (instance) and invoke its methods to consume the Oracle Database adapter. For information about how to generate the WCF client class and helper code for operations that the Microsoft BizTalk Adapter for Oracle Database exposes, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

To create the WCF client, you must specify an endpoint address and a binding. The endpoint address must contain a valid Oracle connection URI, and the binding must be an instance of an Oracle DB Binding (**OracleDBBinding**). For more information about the Oracle connection URI, see [The Oracle Database Connection URI](#). We recommend that you do not specify the user credentials as part of the connection URI. You may instead use the **ClientCredentials** property of the WCF client, as explained in this topic.

You can specify the Oracle DB Binding and the endpoint address in your code or in a configuration file. When you use the Add Adapter Service Reference Visual Studio Plug-in to generate the WCF client class, a configuration file (app.config) is also created for your project. This file contains configuration settings that reflect the binding properties and connection information (except credentials) that you specified when you connected to the Oracle database with the Add Adapter Service Reference Plug-in.

Specifying the Binding and Endpoint Address in Code

The following code shows how to create a WCF client by specifying the binding and endpoint address in code. It is good practice to specify the Oracle credentials by using the **ClientCredentials** property of the WCF client rather than in the connection URI supplied for the endpoint address.

```
// A WCF client that targets the /SCOTT/EMP table is created
// by using a binding object and endpoint address
OracleDBBinding odbBinding = new OracleDBBinding();
EndpointAddress odbAddress = new EndpointAddress("OracleDb://ADAPTER");

SCOTTTableEMPClient empClient = new SCOTTTableEMPClient(odbBinding, odbAddress);

empClient.ClientCredentials.UserName.UserName = "SCOTT";
empClient.ClientCredentials.UserName.Password = "TIGER";

empClient.Open();
```

Specifying the Binding and Endpoint Address in a Configuration File

The following code shows how to create a WCF client by specifying the binding and endpoint address in an app.config file.

```
// A WCF client that targets the /SCOTT/EMP table is created
// by specifying the client endpoint information in app.config
SCOTTTableEMPClient empClient = new SCOTTTableEMPClient("OracleDBBinding_SCOTT.Table.EMP");

empClient.ClientCredentials.UserName.UserName = "SCOTT";
empClient.ClientCredentials.UserName.Password = "TIGER";

empClient.Open();
```

The following XML shows the configuration file created for the EMP table by the Add Adapter Service Reference Plug-in. This file contains the client endpoint configuration referenced in the preceding example.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <oracleDBBinding>
        <binding name="OracleDBBinding" closeTimeout="00:01:00" openTimeout="00:01:00"
receiveTimeout="00:10:00" sendTimeout="00:01:00"
dataFetchSize="65536" metadataPooling="true" statementCachePurge="false" />
      </oracleDBBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

```

statementCacheSize="10" longDatatypeColumnSize="32767" pollingStatement
="""
postPollStatement="" pollingInterval="500" useOracleConnectionPool="fal
se"
minPoolSize="1" maxPoolSize="100" incrPoolSize="5" decrPoolSize="1"
connectionLifetime="0" transactionIsolationLevel="ReadCommitted"
enablePerformanceCounters="false" acceptCredentialsInUri="false"
enableBizTalkCompatibilityMode="false" />
</oracleDBBinding>
</bindings>
<client>
  <endpoint address="oracledb://adapter/" binding="oracleDBBinding"
    bindingConfiguration="OracleDBBinding" contract="SCOTTTableEMP"
    name="OracleDBBinding_SCOTT.Table.EMP" />
</client>
</system.serviceModel>
</configuration>

```

If a project has more than one WCF client, there will be multiple client endpoint entries defined in the configuration file. Each WCF client entry will have a unique name based on its binding configuration and target Oracle database artifact; for example, "OracleDBBinding_SCOTT.Table.EMP". If you connect multiple times to create the WCF clients in your project, multiple binding configuration entries will be created, one for each connection. These binding configuration entries will be named in the following manner: OracleDBBinding1, OracleDBBinding2, and so on. Each client endpoint entry created during a specific connection will reference the binding entry created during that connection.

See Also

Concepts

[Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#)

[The Oracle Database Connection URI](#)

Other Resources

[Developing Applications by Using the WCF Service Model](#)

[Developing Applications by Using the WCF Channel Model](#)

Performing Basic Insert, Update, Delete, and Select Operations by Using the WCF Service Model

The Microsoft BizTalk Adapter for Oracle Database surfaces a set of basic Insert, Update, Delete, and Select operations on Oracle database tables and views. By using these operations, you can perform simple SQL INSERT, UPDATE, SELECT, and DELETE statements qualified by a WHERE clause on a target table or view. To perform more complex operations, for example a SQL SELECT query that uses the JOIN operator, you can use the SQLEXECUTE operation. For more information about the SQLEXECUTE operation, see [Performing a SQLEXECUTE Operation by Using the WCF Service Model](#).

The following table summarizes the basic SQL operations that the Oracle Database adapter surfaces on tables and views. For a complete description of these operations, see [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#).

Operation	Description
Insert	The Insert operation supports multiple record or bulk inserts into the target table or view: <ul style="list-style-type: none"> • A multiple record Insert operation inserts rows into a table or view based on a supplied record set. • A bulk Insert operation inserts rows into a table or view based on a supplied SQL SELECT query and column list. The records that the query returns are inserted into the target table based on the column list.
Select	Performs a SQL SELECT query on the target table based on a supplied list of column names and a filter string that specifies a SQL WHERE clause.
Update	Performs an UPDATE on the target table. The records to be updated are specified by a filter string that specifies a SQL WHERE clause. The values for the update are specified in a template record.
Delete	Performs a DELETE on the target table based on a SQL WHERE clause that is specified in a filter string.

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/ACCOUNTACTIVITY table. A script to generate this table is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

The WCF Client Class

The name of the WCF client generated for the basic SQL operations that the Oracle Database adapter surfaces is based on the name of the table or view, as in the following table.

Oracle Database Artifact	WCF Client Name
Table	[SCHEMA]Table[TABLE_NAME]Client
View	[SCHEMA]View[VIEW_NAME]Client

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = The name of the table; for example, ACCOUNTACTIVITY.

[VIEW_NAME] = The name of the view.

The following table shows the method signatures for the basic SQL operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Insert	long Insert([TABLE_NS].[TABLE_NAME]RECORDINSERT[] RECORDSET, string COLUMN_NAMES, string QUERY);

Select	[TABLE_NS].[TABLE_NAME]RECORDSELECT[] Select(string COLUMN_NAMES, string FILTER);
Update	long Update([TABLE_NS].[TABLE_NAME]RECORDUPDATE RECORDSET, string FILTER);
Delete	Long Delete(string FILTER);

[TABLE_NS] = The name of the table namespace; for example, microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.

[TABLE_NAME] = The name of the table; for example, ACCOUNTACTIVITY.

The record types used by the Insert, Update, and Select operations are all defined in the table or view namespace.

The following code shows the method signatures for a WCF client class generated for the Delete, Insert, Select and Update operations on the /SCOTT/ACCOUNTACTIVITY table.

```
public partial class SCOTTTableACCOUNTACTIVITYClient : System.ServiceModel.ClientBase<SCOTTTableACCOUNTACTIVITY>, SCOTTTableACCOUNTACTIVITY {

    public long Delete(string FILTER);

    public long Insert(microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDINSERT[] RECORDSET, string COLUMN_NAMES, string QUERY);

    public microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDSELECT[] Select(string COLUMN_NAMES, string FILTER);

    public long Update(microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDUPDATE RECORDSET, string FILTER);
}
```

Invoking the Basic SQL Operations

To invoke the basic SQL operations on a table or view by using a WCF client, perform the following steps.

1. Generate a WCF client class for the target table or view. This class should contain methods for the operations that you will invoke on the target artifact.
2. Create an instance of the WCF client class and invoke its methods to perform operations on the table or view.

For more detailed information about how to create a WCF client class and invoke operations on the Oracle Database adapter, see [Overview of Using the WCF Service Model with the Adapter](#).

The Oracle Database adapter executes each operation inside of a transaction on the Oracle database. You can control the isolation level of this transaction by setting the **TransactionIsolationLevel** binding property. For more information about the Oracle Database adapter binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

The following sections provide details about how to invoke each basic SQL operation in your code.

Insert Operation

The following table shows how to set parameters for multiple record Insert and bulk Insert operations.

Insert operation type	RECORDSET	COLUMN_NAMES	QUERY

Multiples records	A collection of INSERTRECORDS that should be inserted into the target.	null	null
Block	null	A comma-delimited list of column names in the target; for example, "TID, ACCOUNT". The column list specifies the columns into which the query results should be placed in each inserted row. The query must return a result set that matches the columns specified in the column list in both number and type.	A SQL SELECT query on a database table or view that returns a result set to insert into the target; for example, "SELECT (TID, ACCOUNT) FROM NEW_TRANSACTIONS WHERE ACCOUNT = 100001". The result set must match the column list in both number and type.

The Insert operation returns the number of records inserted into the target.

◆ Important

In the WCF service model, the record set used in the Insert operation is strongly-typed. You can set the value of a nullable column to **null** in a record to exclude that column from the Insert operation; however, you cannot set the value of a non-nullable column to **null**. This means that in a multiple record Insert operation, you must supply values for all non-nullable columns in each record. In addition, there is no streaming support for the basic SQL operations when you use the WCF service model. If your multiple record Insert operation involves a large record set, this may be an important consideration. For more information, see [Limitations of Invoking the Basic SQL Operations by Using the WCF Service Model](#).

The following code shows a multiple record Insert operation (two records) that targets the ACCOUNTACTIVITY table.

```
// Insert records
using (SCOTTTableACCOUNTACTIVITYClient aaTableClient =
    new SCOTTTableACCOUNTACTIVITYClient("OracleDBBinding_SCOTT.Table.ACCOUNTACTIVITY"))
{
    long recsInserted;

    aaTableClient.ClientCredentials.UserName.UserName = "SCOTT";
    aaTableClient.ClientCredentials.UserName.Password = "TIGER";

    try
    {
        aaTableClient.Open();
    }
    catch (Exception ex)
    {
        // handle exception
        Console.WriteLine("Exception: " + ex.Message);
        throw;
    }

    // Do a multiple record Insert of 2 records for account 100001

    microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDINSERT[] insertRecs =
        new microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDINSERT[2];

        TID__COMPLEX_TYPE tid = new TID__COMPLEX_TYPE();
        tid.InlineValue = "tidSequence.NextVal()";

        ACCOUNT__COMPLEX_TYPE account = new ACCOUNT__COMPLEX_TYPE
        ();

        account.Value = 100001;

        AMOUNT__COMPLEX_TYPE amount = new AMOUNT__COMPLEX_TYPE();
```

```

amount.Value = 400;

TRANSDATE__COMPLEX_TYPE transdate = new TRANSDATE__COMPLEX_TYPE();
transdate.Value = DateTime.Now.Date;

PROCESSED__COMPLEX_TYPE processed = new PROCESSED__COMPLEX_TYPE();
processed.Value = "n";

DESCRIPTION__COMPLEX_TYPE description1 = new DESCRIPTION__COMPLEX_TYPE(
);
description1.Value = "Inserted Record #1";

DESCRIPTION__COMPLEX_TYPE description2 = new DESCRIPTION__COMPLEX_TYPE(
);
description2.Value = "Inserted Record #2";

insertRecs[0] =
    new microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACT
    IVITY.ACCOUNTACTIVITYRECORDINSERT();
insertRecs[0].TID = tid;
insertRecs[0].ACCOUNT = account;
insertRecs[0].AMOUNT = amount;
insertRecs[0].TRANSDATE = transdate;
insertRecs[0].DESCRIPTION = description1;
insertRecs[0].PROCESSED = processed;

insertRecs[1] =
    new microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACT
    IVITY.ACCOUNTACTIVITYRECORDINSERT();
insertRecs[1].TID = tid;
insertRecs[1].ACCOUNT = account;
insertRecs[1].AMOUNT = amount;
insertRecs[1].TRANSDATE = transdate;
insertRecs[1].DESCRIPTION = description2;
insertRecs[1].PROCESSED = processed;

try
{
    recsInserted = aaTableClient.Insert(insertRecs, null, null);
}
catch (Exception ex)
{
    // handle exception
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("Insert Done: {0} records inserted", recsInserted);

```

Select Operation

The following table shows the parameters for the Select operation.

COLUMN_NAMES	FILTER
A comma-delimited list of column names in the target; for example, "TID, ACCOUNT". The column list specifies the columns of the target that should be returned in the result set. Columns not specified in the column list will be set to their .NET default values in the returned record set. For nullable columns, this value is null .	The contents of a SQL WHERE clause that specifies the target rows of the query; for example, "DESCRIPTION = 'Insert Record #1'". You can set this parameter to null to return all rows of the target.

The Select operation returns a strongly-typed record set based on the row type of the target.

◆ Important

There is no streaming support for the basic SQL operations when you use the WCF service model. If your query returns a large record set, you might be able to improve performance by using the WCF channel model. For more information, see [Limitations of Invoking the Basic SQL Operations by Using the WCF Service Model](#).

The following code shows a Select operation that targets the ACCOUNTACTIVITY table. The returned records are written to the console.

```
// Declare a variable to hold the result set
microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDS
ELECT[] selectRecords;

// Select all records and write them to the console
try
{
    selectRecords = aaTableClient.Select("*", null);
}
catch (Exception ex)
{
    // handle exception
}

Console.WriteLine("ACCOUNTACTIVITY before any operations");
for (int i = 0; i < selectRecords.Length; i++)
{
    Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}", selectRecords[i].TID,
        selectRecords[i].ACCOUNT,
        selectRecords[i].AMOUNT,
        selectRecords[i].TRANSDATE,
        selectRecords[i].DESCRIPTION);
}
```

Note

This code omits steps to create, configure, and open the WCF client instance. For an example that includes these steps, see [Insert Operation](#).

Update Operation

The following table shows the parameters for the Update operation.

RECORDSET	FILTER
A strongly-typed template record based on the row type of the target. The template record specifies the update values for the target rows. For nullable row columns, you can specify a null value to indicate that the column should not be updated in the target rows.	The contents of a SQL WHERE clause that specifies the rows to be updated in the target. For example, "DESCRIPTION= 'Inserted Record #1'".

The Update operation returns the number of rows deleted from the target.

Important

In the WCF service model, the template record used in the Update operation is strongly-typed. If a column is nullable, you can omit the column from the Update operation by setting its value to **null** in the template record; however, if a column is not nullable, then you must set its value in the template record. For example, if a column is a primary key, it must contain a value. For more information, see [Limitations of Invoking the Basic SQL Operations by Using the WCF Service Model](#).

The following code shows an Update operation that targets the ACCOUNTACTIVITY table.

```
long recsUpdated;

...

// Create updated template. The TID, TIME, AMOUNT, and DESCRIPTION fields will be updated
```

```

microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDU
PDATE updateRecord =
    new microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVIT
YRECORDUPDATE();
    updateRecord.TID = tidSequence.NextVal();
    updateRecord.ACCOUNT = null;
    updateRecord.AMOUNT = 300;
    updateRecord.TRANSDATE = DateTime.Now.Date;
    updateRecord.DESCRPTION = "Updated Record #2";
    updateRecord.PROCESSED = null;

// Set filter string to specify the target record by using the DESCRIPTION field
string filter = "DESCRIPTION = 'Inserted Record #2'";

try
{
    recsUpdated = aaTableClient.Update(updateRecord, filter);
}
catch (Exception ex)
{
    // handle exception
    ...
}

Console.WriteLine("{0} records updated", recsUpdated);

```

Note

This code omits steps to create, configure, and open the WCF client instance. For an example that includes these steps, see [Insert Operation](#).

Delete Operation

The following table shows the parameters for the Delete operation.

FILTER

The contents of a SQL WHERE clause that specifies the rows to be deleted from the target. For example, "DESCRIPTION= 'Inserted Record #1'".

The Delete operation returns the number of rows deleted from the target. The following code shows a Delete operation that targets the ACCOUNTACTIVITY table.

```

// Set filter string equal to the DESCRIPTION field of the target record
string filter = "DESCRIPTION = 'Inserted Record #1'";

try
{
    recsDeleted = aaTableClient.Delete(filter);
}
catch (Exception ex)
{
    // handle exception
    ...
}

Console.WriteLine("{0} records deleted", recsDeleted);

```

Note

This code omits steps to create, configure, and open the WCF client instance. For an example that includes these steps, see the [Insert Operation](#).

Limitations of Invoking the Basic SQL Operations by Using the WCF Service Model

The following limitations exist when you invoke the basic SQL operations by using a WCF client:

- **Insert operation.** The record set used in a multiple record Insert operation is strongly-typed and therefore includes all row columns. The Oracle Database adapter interprets a null value in a record to mean that the column should be excluded from the Insert operation; however, non-nullable columns cannot be excluded because you cannot set them to a null value. Therefore, you must specify values for non-nullable columns when you perform a multiple record Insert operation.
- **Insert operation.** The Oracle Database adapter interprets a **DBNull** value in a nullable data column to mean that the column should be excluded from a multiple record Insert operation. This means that you cannot set a nullable column to **DBNull** on the Oracle database in a multiple record Insert operation.
- **Insert operation.** There is no streaming support for multiple record insert operations that involve a large record set.
- **Update operation.** The template record used in an Update operation is strongly-typed and therefore includes all row columns. The Oracle Database adapter interprets a null value in this record to mean that the column should be excluded from the Update operation; however, non-nullable columns cannot be excluded because you cannot set them to a null value. Therefore, you must specify values for non-nullable columns when you perform an Update operation.
- **Update operation.** The Oracle Database adapter interprets a **DBNull** value in a nullable data column in the template record to mean that the column should be excluded from the operation. This means that you cannot set a nullable column to **DBNull** on the Oracle database by using the Update operation.
- **Select operation.** There is no streaming support for SELECT queries that return a large record set.

For scenarios where these limitations present challenges, you can invoke the operation by using the WCF channel model because:

- By using the WCF channel model, you can exclude specific data columns from Update and Insert operations.
- The WCF channel model provides node-level streaming support for the basic SQL operations that the Oracle Database adapter exposes.

For more information about using the WCF channel model with the Oracle Database adapter, see [Developing Applications by Using the WCF Channel Model](#).

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing Operations on Tables with Large Object Types by Using the WCF Service Model

This section contains information about how to invoke the ReadLOB and UpdateLOB operations from the WCF service model. The ReadLOB and UpdateLOB operations are surfaced for tables and views that contain LOB columns; that is columns that are used to store Oracle large object (LOB) data. For an overview of the Oracle LOB data types supported by the Microsoft BizTalk Adapter for Oracle Database and of the ReadLOB and UpdateLOB operations, see [Operations on Tables and Views That Contain LOB Data](#).

Important

LOB data columns can contain large amounts of data—up to 4 gigabytes (GB). A significant limitation of using a WCF client to operate on LOB columns is that the WCF service model only supports data streaming on the ReadLOB operation, not on the UpdateLOB operation. This is because WCF requires that for streaming to work from service model, the parameter to be streamed must be the only parameter in its direction. The UpdateLOB operation has two other IN parameters (a column name and row filter) in addition to the LOB data; for this reason, streaming is not supported on it in the WCF service model. Therefore, if you are updating a LOB column with a large amount of data, you might want to use the WCF channel model. For more information on how to use the WCF channel model to stream LOB data using the UpdateLOB operation, see [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#).

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/CUSTOMER table. This table contains a BLOB column named PHOTO. A script to generate this table is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

The WCF Client Class

The following example shows the method signatures for a WCF client class generated for the ReadLOB and UpdateLOB operations on the /SCOTT/CUSTOMER table.

```
public partial class SCOTTTableCUSTOMERClient : System.ServiceModel.ClientBase<SCOTTTableCUSTOMER>,
    SCOTTTableCUSTOMER
{
    public System.IO.Stream ReadLOB(string LOB_COLUMN, string FILTER);

    public void UpdateLOB(string LOB_COLUMN, string FILTER, byte[] Stream);
}
```

Note

Note that **ReadLOB** returns a data stream, but that **UpdateLOB** does not operate on a stream.

Invoking the ReadLOB and UpdateLOB Operations

Both the **ReadLOB** and **UpdateLOB** methods can operate only on a single LOB column in a single database row. You set the following parameters to identify the target column/row.

Parameter	Definition
LOB_COLUMN	The name of the target column within the row identified by the FILTER parameter; for example, "PHOTO".

FILTER	<p>The contents of a SQL SELECT statement WHERE clause that specifies the target row; for example, "NAME='Kim Ralls'". The filter must specify one and only one row. If the filter matches more than one row:</p> <ul style="list-style-type: none"> • ReadLOB returns LOB data for the first matching row. • UpdateLOB throws an exception.
--------	--

Note
The stream returned by **ReadLOB** does not support **Seek**. This means that properties such as **Length** are not supported, either.

Caution
The **UpdateLOB** operation must be performed within a transaction scope. Also, the **UseAmbientTransaction** binding property must be set to **true** before performing the **UpdateLOB** operation.

The following code shows how to use a WCF client to update the BLOB PHOTO column in the /SCOTT/CUSTOMER table from a file and read the new column data back to a file. You can find a full sample in the SDK samples. For more information about the SDK samples, see [Samples](#).

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Transaction;

// Include for file streaming
using System.IO;

// Add WCF, WCF LOB Adapter SDK, and Oracle Database adapter namespaces
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

// Include this namespace for the WCF channel model
using System.ServiceModel.Channels;

// Include this namespace for the WCF LOB Adapter SDK and Oracle Database adapter exceptions
using Microsoft.ServiceModel.Channels.Common;

using CustomerTablens = microsoft.lobservices.oracledb._2007._03;

namespace OracleLobOpsSnippetSM
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                OracleDBBinding binding = new OracleDBBinding();
                binding.UseAmbientTransaction = true; //set this to true for UpdateLOB operation

                EndpointAddress endpointAddress = new EndpointAddress("oracleDB://ADAPTER");

                using (SCOTTTableCUSTOMERClient customerTableClient =
                    new SCOTTTableCUSTOMERClient(binding, endpointAddress))
            {

```


Invoking Functions and Procedures by Using the WCF Service Model

The Microsoft BizTalk Adapter for Oracle Database surfaces procedures, functions, and packages as operations. In the WCF service model these operations are represented as methods on a WCF client. The WCF service model and the Oracle Database adapter:

- **Support functions.** The RETURN value of the Oracle function is surfaced as the return value of the WCF client method. Oracle parameters are surfaced as parameters (with the appropriate direction as defined below) to the WCF client method.
- **Support procedures.** The first OUT parameter of the Oracle procedure is surfaced as the return value of the WCF client method. All other Oracle parameters are surfaced as parameters (with the appropriate direction as defined below) to the WCF client method.
- **Support Oracle packages.** The name of the operation and the namespace of its parameter types are qualified by the package name.
- **Support overloaded functions and procedures.**
- **Support IN, OUT and IN OUT parameters for basic Oracle data types for both procedures and functions.** OUT parameters are surfaced as **out** parameters on the WCF client method and IN OUT parameters are surfaced as **ref** parameters.
- **Support IN, OUT, and IN OUT REF CURSOR parameters for procedures and functions, as well as function RETURN values.** For more information, see [Performing Operations Using REF CURSORS in the WCF Service Model](#).
- **Support IN, OUT, and IN OUT RECORD type parameters for procedures and functions, as well as function RETURN values.** For more information, see [Performing Operations Using RECORD Types in the WCF Service Model](#).

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT overloaded procedure. This procedure reads a record from the SCOTT/ACCOUNT table based on either an account ID or an account name. A script to generate this procedure and table is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

The WCF Client Class

The following table shows the name of the WCF client and the method generated for procedures, functions and packages that the Oracle Database adapter surfaces. Unless a function or procedure is overloaded, a single WCF client is used to invoke all of the functions in a schema, all of the procedures in a schema, or all of the functions and procedures in a package.

Oracle Artifact	WCF Client Operation Name	Example
Procedure	[SCHEMA]ProcedureClient.[PROC_NAME]	SCOTTProcedureClient.MYPROC
Function	[SCHEMA]FunctionClient.[FUNC_NAME]	SCOTTProcedureClient.MYFUNC
Package (procedure or function)	[SCHEMA]Package[PACKAGE_NAME]Client.[PROC_NAME or FUNC_NAME]	SCOTTPackageMYPACKAGEClient.MYPROC

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[PROC_NAME] = The name of an Oracle procedure; for example, MYPROC.

[FUNC_NAME] = The name of an Oracle function; for example, MYFUNC.

[PACKAGE_NAME] = The name of an Oracle package.

The Oracle Database adapter represents Oracle RECORD type parameters and return values as well as the result sets returned by REF CURSOR parameters as complex XML types that contain the row data (or fields) of an Oracle record. In the WCF service model, each of these XML types is represented as a .NET class; the properties of the class represent the fields of the RECORD type or REF CURSOR result set. Oracle RECORD types are always represented as strongly-typed .NET classes. A REF CURSOR result set, however, can be represented as either strongly-typed or weakly-typed records based on whether the REF CURSOR itself is declared as strongly-typed or weakly-typed. The classes that represent REF CURSOR or RECORD type parameters (or return values) are generated in a unique namespace based on the procedure, function, or package. The following table shows these namespaces.

Oracle Artifact	Namespace	Example
Procedure	[BASE_NS]. [SCHEMA].Procedure.[PROC_NAME]	microsoft.lobservices.oracledb_2007_03.SCOTT.Procedure.MYPROC
Function	[BASE_NS]. [SCHEMA].Function.[FUNCTION_NAME]	microsoft.lobservices.oracledb_2007_03.SCOTT.Function.MYFUNC
Package (Procedure)	[BASE_NS]. [SCHEMA].Package.[PACKAGE_NAME].[PROC_NAME]	microsoft.lobservices.oracledb_2007_03.SCOTT.Package.MYPACKAGE.MYPROC
Package (Function)	[BASE_NS]. [SCHEMA].Package.[PACKAGE_NAME].[FUNCTION_NAME]	microsoft.lobservices.oracledb_2007_03.SCOTT.Package.MYPACKAGE.MYFUNC
Generic Record Set (weakly-typed)	[BASE_NS]	microsoft.lobservices.oracledb_2007_03

[BASE_NS] = The base adapter namespace; microsoft.lobservices.oracledb_2007_03.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[PROC_NAME] = The name of an Oracle procedure; for example, MYPROC.

[FUNCTION_NAME] = The name of an Oracle function; for example MYFUNC.

[PACKAGE_NAME] = The name of an Oracle package.

For information about how these namespaces are used for RECORD parameters, see [Performing Operations Using RECORD Types in the WCF Service Model](#). For information about how these namespaces are used for REF CURSOR parameters, see [Performing Operations Using REF CURSORS in the WCF Service Model](#).

In general, the Oracle parameters and return values are mapped as follows in the WCF client method:

- Oracle IN parameters are mapped to .NET (input) parameters.
- Oracle OUT parameters are mapped to .NET **out** parameters.
- Oracle IN OUT parameters are mapped to .NET **ref** parameters.
- Function RETURN values are mapped to the method return value.

However, two important exceptions exist:

- Oracle IN OUT REF CURSOR parameters are split into an input string and an output (**out**) record set. This is because the Oracle Database adapter represents IN REF CURSOR parameters as strings and OUT REF CURSOR parameters as complex types (record sets), these cannot be combined into a single parameter.
- The first OUT parameter in an Oracle procedure is mapped to the return value of the WCF client method. This is standard WCF behavior.

The following example shows part of a simple Oracle procedure (loaded in the SCOTT schema) and the signature of the WCF client method that is generated to invoke it. The Oracle procedure has three IN parameters, three IN OUT parameters, and

three OUT parameters; however, the WCF client method does not map a parameter for the first OUT parameter. Instead it is mapped to the method return value.

```
CREATE or REPLACE PROCEDURE Sample_Procedure
(
  INNUMBER      IN      NUMBER,
  INVARCHAR     IN      VARCHAR2,
  INDATE       IN      DATE,
  INOUTNUMBER  IN OUT  NUMBER,
  INOUTVARCHAR IN OUT  VARCHAR,
  INOUTDATE    IN OUT  DATE,
  OUTNUMBER    OUT     NUMBER,
  OUTVARCHAR   OUT     VARCHAR2,
  OUTDATE      OUT     DATE
) AS
BEGIN
    ...

END;
/
```

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
public partial class SCOTTProcedureClient : System.ServiceModel.ClientBase<SCOTTProcedure>,
SCOTTProcedure {
```

```
    public System.Nullable<decimal> SAMPLE_PROCEDURE
    (
        System.Nullable<decimal> INNUMBER,
        string INVARCHAR,
        System.Nullable<System.DateTime> INDATE,
        ref System.Nullable<decimal> INOUTNUMBER,
        ref string INOUTVARCHAR,
        ref System.Nullable<System.DateTime> INOUTDATE,
        out string OUTVARCHAR,
        out System.Nullable<System.DateTime> OUTDATE
    );
}
```

Support for Overloaded Procedures, Functions and Packages

The Oracle Database adapter supports overloaded procedures, functions, and packages by appending a unique string to the node ID and the namespace that it surfaces for each overloaded artifact. This string is "overload1" for the first overload, "overload2" for the next overload, and so on.

In the WCF service model each overloaded procedure or function is represented by a unique WCF client. This is different from the non-overloaded case in which all of the functions in a SCHEMA, all of the procedures in a SCHEMA, or all of the procedures and functions in a PACKAGE are invoked by the same WCF client. The following table shows the WCF client name and method generated for overloaded procedures, functions, and packages.

Oracle Artifact	WCF Client Name	Example
Overloaded Package (Procedure)	[SCHEMA]Package[PACKAGE_NAME][PROC_NAME]][OVERLOAD_ID]Client.[PROC_NAME]	SCOTTPackageMYPACKAGEMYPROCoverload1Client.MYPROC
Overloaded Package (Function)	[SCHEMA]Package[PACKAGE_NAME][FUNC_NAME]][OVERLOAD_ID]Client.[FUNC_NAME]	SCOTTPackageMYPACKAGEMYFUNCoverload1Client.MYFUNC

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[PROC_NAME] = The name of an Oracle procedure; for example, MYPROC.

[FUNC_NAME] = The name of an Oracle function; for example MYFUNC.

[PACKAGE_NAME] = The name of an Oracle package.

[OVERLOAD_ID] = The unique string that identifies the overloaded artifact; "overload1", "overload2", and so on.

The following table shows the namespace generated for overloaded procedures, functions, and packages.

Oracle Artifact	Namespace	Example
Package (Procedure)	[BASE_NS]. [SCHEMA].Package.[PACKAGE_NAME].[PROC_NAME] [OVERLOAD_ID]	microsoft.lobservices.oracledb_2007_03.SCOTT.Package.MYPACKAGE.MYPROC.overload1
Package (Function)	[BASE_NS]. [SCHEMA].Package.[PACKAGE_NAME].[FUNC_NAME].[OVERLOAD_ID]	microsoft.lobservices.oracledb_2007_03.SCOTT.Package.MYPACKAGE.MYFUNC.overload1
Generic Record Set (weakly-typed)	[BASE_NS]	microsoft.lobservices.oracledb_2007_03

[BASE_NS] = The base adapter namespace; microsoft.lobservices.oracledb_2007_03.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[PROC_NAME] = The name of an Oracle procedure; for example, MYPROC.

[FUNC_NAME] = The name of an Oracle function; for example MYFUNC.

[PACKAGE_NAME] = The name of an Oracle package.

[OVERLOAD_ID] = The unique string that identifies the overloaded artifact; "overload1", "overload2", and so on. The numeric value in the string is the overload ID for the artifact maintained by the Oracle database.

The following example shows the WCF clients and the method signatures generated for the overloaded GET_ACCOUNT procedure in the ACCOUNT_PKG package. (The Oracle declarations are included.) This example shows how a unique WCF client is generated for each overload and how the method generated for each client returns a record set in a unique namespace.

```
/* Procedure that takes account ID and returns record for existing account in the ACCOUNT table */
PROCEDURE get_account(aid IN account.acctid%TYPE, acct OUT account%ROWTYPE) ;

/* Procedure that takes account name and returns record for existing account in the ACCOUNT table */
PROCEDURE get_account(aname IN account.name%TYPE, acct OUT account%ROWTYPE) ;

[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
public partial class SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload1Client : System.ServiceModel.ClientBase<SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload1>, SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload1 {

    public microsoft.lobservices.oracledb_2007_03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNT.Overload1.ACCTRECORD GET_ACCOUNT(System.Nullable<decimal> AID);
}

[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
public partial class SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload2Client : System.ServiceModel.ClientBase<SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload2>, SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload2 {

    public microsoft.lobservices.oracledb_2007_03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNT.Overload2.ACCTRECORD GET_ACCOUNT(string ANAME);
}
```

Invoking Functions and Procedures

To invoke a function or a procedure by using a WCF client, perform the following steps.

1. Generate a WCF client class for the target function, procedure, or package. This class should contain methods for the operations that you will invoke on the target artifact.

Note

In the Add Adapter Service Reference Visual Studio Plug-in, overloaded functions and procedures appear in the **Available categories and operations** box as [NAME].1, [NAME].2, [NAME].3, and so on, where [NAME] is the name of the overloaded artifact and the numeric value is the overload ID on the Oracle database.

2. Create an instance of the WCF client class and call its methods to invoke the function or procedure.

For more detailed information about how to create a WCF client class and invoke operations on the Oracle Database adapter, see [Overview of Using the WCF Service Model with the Adapter](#).

The Oracle Database adapter executes each operation inside of a transaction on the Oracle database.

Important

The classes that represent REF CURSOR and RECORD type parameters or return values in functions or procedures (and packages) are declared in a unique namespace for each function or procedure. This means, for example, that a PACKAGE REF CURSOR type that is used as a return value in two different functions will be declared in a unique namespace for each WCF client method. You must either declare separate variables to hold these different return values or appropriately cast the variable when you invoke one of the WCF client methods.

The following example demonstrates calling the overloaded /SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT procedure to get account records from the /SCOTT/ACCOUNT table. First a new record is created by calling the /SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT procedure. Then the new record is read back twice by calling different overloads of GET_ACCOUNT. This example uses three WCF clients, one for the CREATE_ACCOUNT procedure and one each for the GET_ACCOUNT overloads. Aliases are used to distinguish between namespaces used for the return value of GET_ACCOUNT. A full sample is available in the SDK samples. For more information about the SDK samples, see [Samples](#).

```
using System;
using System.Collections.Generic;
using System.Text;

// Add WCF, WCF Adapter LOB SDK, and Oracle Database adapter namespaces
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

// Include this namespace for WCF Adapter LOB SDK and Oracle Database adapter exceptions
using Microsoft.ServiceModel.Channels.Common;

// Alias client namespaces to shorten declarations of "shared" types
using CREATE_ACCOUNTns = microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG
.CREATE_ACCOUNT;
using GET_ACCOUNT_BY_IDns = microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_
PKG.GET_ACCOUNT.overload1;
using GET_ACCOUNT_BY_NAMEns = microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUN
T_PKG.GET_ACCOUNT.overload2;

// This sample demonstrates calling overloaded packaged procedures on Oracle
// First a new account is created by calling CREATE_ACCOUNT which takes two record paramete
rs
// Then the information for the new account is returned by calling an overloaded procedure
GET_ACCOUNT
// The first overload returns the account information by account ID
// The second overload returns the account information by account name
// Notice that different clients (and namespaces) are created for overloaded procedures and
```

```

functions
namespace OracleOverloadsSM
{
    class Program
    {
        static void Main(string[] args)
        {
            decimal acctId;
            string newAccountName = "Paula Bento";

            Console.WriteLine("Creating clients");
            // Create Client for CREATE_ACCOUNT Function
            SCOTTPackageACCOUNT_PKGClient createAccountClient =
                new SCOTTPackageACCOUNT_PKGClient("OracleDBBinding_SCOTT.Package.ACCOUNT_PK
G");
            // NOTE: user name and password are case-sensitive
            createAccountClient.ClientCredentials.UserName.UserName = "SCOTT";
            createAccountClient.ClientCredentials.UserName.Password = "TIGER";

            // Create Client for GET_ACCOUNT Overload 1 -- takes ACCOUNT ID parameter
            SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload1Client getAccountByIdClient =
                new SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload1Client("OracleDBBinding_SCOT
T.Package.ACCOUNT_PKG.GET_ACCOUNT.overload1");
            // NOTE: user name and password are case-sensitive
            getAccountByIdClient.ClientCredentials.UserName.UserName = "SCOTT";
            getAccountByIdClient.ClientCredentials.UserName.Password = "TIGER";

            // Create Client for GET_ACCOUNT Overload 2 --takes ACCOUNT NAME parameter
            // NOTE: this client can be created from configuration; detail provided here
            // for demonstration
            OracleDBBinding overload2Binding = new OracleDBBinding();
            EndpointAddress overload2EndpointAddress = new EndpointAddress("oracleDB://ADAP
TER");
            SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload2Client getAccountByNameClient =
                new SCOTTPackageACCOUNT_PKGGET_ACCOUNTOverload2Client(overload2Binding, ove
rload2EndpointAddress);
            // NOTE: user name and password are case-sensitive
            getAccountByNameClient.ClientCredentials.UserName.UserName = "SCOTT";
            getAccountByNameClient.ClientCredentials.UserName.Password = "TIGER";

            try
            {
                Console.WriteLine("Opening clients -- please wait");
                // Open clients
                createAccountClient.Open();
                getAccountByIdClient.Open();
                getAccountByNameClient.Open();

                Console.WriteLine("Creating new account");
                // Create an account record
                // NOTE: ACCTRECORD is defined in all three namespaces so specify the defin
ition
                // that corresponds to the client.
                CREATE_ACCOUNTns.ACCTRECORD acctRec = new CREATE_ACCOUNTns.ACCTRECORD();

                // Set any value for ACCTID -- new account ID is returned by CREATE_ACCOUNT
                acctRec.ACCTID = 0;
                acctRec.NAME = newAccountName;
                acctRec.BALANCE = 10537;

                // Create address record
                CREATE_ACCOUNTns.ACCOUNT_PKGADDRESS_REC_TYPERECORD addrRec = new CREATE_ACC
OUNTns.ACCOUNT_PKGADDRESS_REC_TYPERECORD();
                addrRec.STREET = "456 Valley Rd";
                addrRec.CITY = "New York";
                addrRec.STATE = "NY";
            }
        }
    }
}

```

```

        // Create account
        acctId = (decimal)createAccountClient.CREATE_ACCOUNT(acctRec, addrRec);
        Console.WriteLine("New Account Created: AccountId = {0}, Name = {1}, Balance = {2:C}",
            acctId, acctRec.NAME, acctRec.BALANCE);

        /* Get new account by Id */
        GET_ACCOUNT_BY_IDns.ACCTRECORD acctById = getAccountByIdClient.GET_ACCOUNT(
            acctId);
        Console.WriteLine("Account Returned by Id: AccountId={0}, Name={1}, Balance = {2:C}",
            acctById.ACCTID, acctById.NAME, acctById.BALANCE);

        /* Get new account by Name */
        GET_ACCOUNT_BY_NAMEns.ACCTRECORD acctByName = getAccountByNameClient.GET_ACCOUNT(
            newAccountName);
        Console.WriteLine("Account Returned by Name: AccountId={0}, Name={1}, Balance={2:C}",
            acctByName.ACCTID, acctByName.NAME, acctByName.BALANCE);

        Console.WriteLine("Hit <RETURN> to finish");
        Console.ReadLine();
    }
    catch (TargetSystemException tex)
    {
        Console.WriteLine("Exception occurred on the Oracle Database");
        Console.WriteLine(tex.InnerException.Message);
    }
    catch (ConnectionException cex)
    {
        Console.WriteLine("Exception occurred connecting to the Oracle Database");
        Console.WriteLine(cex.InnerException.Message);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception is: " + ex.Message);
        if (ex.InnerException != null)
        {
            Console.WriteLine("Inner Exception is: " + ex.InnerException.Message);
        }
        throw ex;
    }
    finally
    {
        // Close all the clients
        createAccountClient.Close();
        getAccountByIdClient.Close();
        getAccountByNameClient.Close();
    }
}
}
}

```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing Operations Using REF CURSORS in the WCF Service Model

A REF CURSOR is an Oracle PL/SQL data type that represents a pointer to a result set in the Oracle database. The Microsoft BizTalk Adapter for Oracle Database supports REF CURSOR parameters in procedures, functions, and packages. REF CURSOR parameters can be strongly-typed or weakly-typed depending on how they are declared in the procedure or function. For a detailed explanation of how REF CURSOR parameters are represented by the Oracle Database adapter, see [Message Schemas for REF CURSORS](#). The following table summarizes how REF CURSOR parameters are represented in the WCF service model.

Parameter Direction	Strongly-typed REF CURSOR	Weakly-typed REF CURSOR
IN	string [PARAM_NAME]	Same as strongly-typed
	String that contains a PL/SQL block. The PL/SQL block must return an opened REF CURSOR either by executing an "OPEN FOR SELECT" statement or by invoking a function or procedure. A question mark (?) indicates the position of the REF CURSOR that returns the parameter. For example, "BEGIN OPEN ? FOR SELECT * FROM MY_TABLE; END", or "BEGIN MY_PROC(PARM1, ?, PARM2); END;".	
OUT	out [PROC_NS].[PARAM_NAME]RECORD[] [PARAM_NAME]	out [GENERIC_NS].GenRecordRow[] [PARAM_NAME]
	A strongly-typed record set.	A weakly-typed generic record set.
IN OUT	IN OUT REF CURSOR parameters are split into an IN and an OUT parameter. The IN parameter is appended with "_IN" in the method signature to distinguish it from the OUT parameter. The OUT parameter is represented by a strongly-typed record set.	IN OUT REF CURSOR parameters are split into an IN and an OUT parameter. The IN parameter is appended with "_IN" to distinguish it from the OUT parameter. The OUT parameter is represented by a weakly-typed record set.
	string [PARAM_NAME]_IN	string [PARAM_NAME]_IN
	out [PROC_NS].[PARAM_NAME]RECORD[] [PARAM_NAME]	out [GENERIC_NS].GenRecordRow[] [PARAM_NAME]

[PARAM_NAME] = the name of the parameter in the function or procedure definition on the Oracle database; for example, MYREFCURSOR.

[PROC_NS] = The unique namespace generated to contain parameters of the package, procedure, or function; for example, "microsoft.lobservices.oracledb_2007_03.SCOTT.Package.ACCOUNT_PKG.GET_ACTIVITY".

[GENERIC_NS] = The namespace in which the generic record set is defined, "microsoft.lobservices.oracledb_2007_03".

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/Package/ACCOUNT_PKG Oracle PACKAGE. The following procedure is used from ACCOUNT_PKG:

```
PROCEDURE get_activity(inrecs IN SYS_REFCURSOR, status OUT NUMBER, inoutrecs IN OUT activit
y_ref_type, outrecs OUT SYS_REFCURSOR);
```

A script to generate this package is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

REF CURSOR Parameters in the WCF Service Model

The following examples show the classes and WCF client generated for the /SCOTT/Package/ACCOUNT_PKG/GET_ACTIVITY procedure. This procedure has weakly-typed IN and OUT REF CURSOR parameters and a strongly-typed IN OUT REF CURSOR parameter.

Here is the signature of the method that is generated in the WCF client to invoke GET_ACTIVITY.

```
public System.Nullable<decimal> GET_ACTIVITY(string INRECS, string INOUTRECS_IN, out micros
oft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACTIVITY.INOUTRECSRECORD[]
INOUTRECS, out microsoft.lobservices.oracledb._2007._03.GenRecordRow[] OUTRECS);
```

In the **GET_ACTIVITY** method, the IN OUT parameter INOUTRECS is split into two parameters:

- INOUTRECS_IN is a string that represents an IN REF CURSOR parameter.
- INOUTRECS is a strongly-typed record set that represents an OUT REF CURSOR parameter.

The weakly-typed OUT parameter, OUTRECS, is represented as a generic record set. The weakly-typed IN parameter, INRECS, is represented as a string.

Strongly-Typed OUT REF CURSOR Parameters

Strongly-typed OUT (or IN OUT) REF CURSOR parameters are generated in a unique namespace based on the SCHEMA, PACKAGE, and name of the procedure or function in which they are used. For the /SCOTT/Package/ACCOUNT_PKG/GET_ACTIVITY procedure, this namespace is `microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACTIVITY`. The class name is formed by appending the name of the parameter with "RECORD" and the class is composed of properties that represent the Oracle fields. The following shows a part of the class that represents the strongly-typed records generated for the INOUTRECS REF CURSOR parameter.

```
namespace microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACTIVITY {
    using System.Runtime.Serialization;

    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.
    0")]
    [System.Runtime.Serialization.DataContractAttribute()]
    public partial class INOUTRECSRECORD : object, System.Runtime.Serialization.IExtensible
    DataObject {
        ...

        private System.Nullable<decimal> TIDField;

        ...

        [System.Runtime.Serialization.DataMemberAttribute()]
        public System.Nullable<decimal> TID {
            get {
                return this.TIDField;
            }
            set {
                this.TIDField = value;
            }
        }

        ...
    }
}
```

```
}  
}
```

Weakly-Typed OUT REF CURSOR Parameters

Weakly-typed OUT (or IN OUT) REF CURSOR parameters are represented by the generic record class. The generic record set is always generated in the same namespace and with the same class name regardless of the function or procedure. The following code shows the generic record class, **microsoft.lobservices.oracledb._2007._03.GenRecordRow**, which represents the records for the OUTRECS OUT SYS_REFCURSOR parameter (weakly-typed).

```
namespace microsoft.lobservices.oracledb._2007._03 {  
    using System.Runtime.Serialization;  
  
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.  
0")]  
    [System.Runtime.Serialization.DataContractAttribute()]  
    public partial class GenRecordRow : object, System.Runtime.Serialization.IExtensibleDat  
aObject {  
  
        private System.Runtime.Serialization.ExtensionDataObject extensionDataField;  
  
        private microsoft.lobservices.oracledb._2007._03.GenRecordColumn[] GenRecordColumnF  
ield;  
  
        public System.Runtime.Serialization.ExtensionDataObject ExtensionData {  
            get {  
                return this.extensionDataField;  
            }  
            set {  
                this.extensionDataField = value;  
            }  
        }  
  
        [System.Runtime.Serialization.DataMemberAttribute()]  
        public microsoft.lobservices.oracledb._2007._03.GenRecordColumn[] GenRecordColumn {  
            get {  
                return this.GenRecordColumnField;  
            }  
            set {  
                this.GenRecordColumnField = value;  
            }  
        }  
    }  
  
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.  
0")]  
    [System.Runtime.Serialization.DataContractAttribute()]  
    public partial class GenRecordColumn : object, System.Runtime.Serialization.IExtensible  
DataObject {  
  
        private System.Runtime.Serialization.ExtensionDataObject extensionDataField;  
  
        private string ColumnNameField;  
  
        private string ColumnValueField;  
  
        private string ColumnTypeField;  
  
        public System.Runtime.Serialization.ExtensionDataObject ExtensionData {  
            get {  
                return this.extensionDataField;  
            }  
            set {  
                this.extensionDataField = value;  
            }  
        }  
    }  
}
```

```

    }

[System.Runtime.Serialization.DataMemberAttribute(IsRequired=true, EmitDefaultValue
=false)]
public string ColumnName {
    get {
        return this.ColumnNameField;
    }
    set {
        this.ColumnNameField = value;
    }
}

[System.Runtime.Serialization.DataMemberAttribute(IsRequired=true)]
public string ColumnValue {
    get {
        return this.ColumnValueField;
    }
    set {
        this.ColumnValueField = value;
    }
}

[System.Runtime.Serialization.DataMemberAttribute(IsRequired=true, EmitDefaultValue
=false, Order=2)]
public string ColumnType {
    get {
        return this.ColumnTypeField;
    }
    set {
        this.ColumnTypeField = value;
    }
}
}
}
}

```

Using REF CURSOR Parameters with a WCF Client

To invoke a procedure or function with REF CURSOR parameters by using a WCF client, you do the following:

1. Pass a string for each IN or IN OUT REF CURSOR parameter that contains the PL/SQL block to open the REF CURSOR. This block can either execute an OPEN FOR SELECT statement or invoke a function or procedure that returns an opened REF CURSOR in an OUT parameter.
2. When the procedure or function returns, operate on the data in the record sets returned for any OUT or IN OUT REF CURSOR parameters. The record set will be a generic record set for weakly-typed REF CURSOR parameters or a strongly-typed record set for strongly-typed REF CURSOR parameters.

For more information about how to invoke procedures and functions by using the WCF service model, see [Invoking Functions and Procedures by Using the WCF Service Model](#).

The following example calls the GET_ACTIVITY procedure. It demonstrates both ways of specifying an IN REF CURSOR parameter:

- For the IN REF CURSOR parameter, an OPEN FOR SELECT statement is specified to return activity for ACCOUNT 100001.
- For the IN OUT REF CURSOR parameter, the /SCOTT/Package/ACCOUNT_PKG/GET_ALL_ACTIVITY procedure is invoked. This procedure opens a REF CURSOR that contains all of the activity in the ACCOUNTACTIVITY table and returns it as an OUT parameter.

The example also demonstrates how to read data from the record set returned for both strongly-typed and weakly-typed REF CURSOR parameters.

```
using System;
```

```

using System.Collections.Generic;
using System.Text;

// Add WCF, WCF LOB Adapter SDK, and Oracle Database adapter namespaces
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

// Include this namespace for WCF LOB Adapter SDK and Oracle Database adapter exceptions
using Microsoft.ServiceModel.Channels.Common;

// namespaces for strongly-typed and weakly typed REF CURSOR records
using GET_ACTIVITYns = microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.G
ET_ACTIVITY;
using GENERICns = microsoft.lobservices.oracledb._2007._03;

// In this sample, INRECS is opened by using an OPEN FOR statement, and
// INOUTRECS_IN is opened by calling the GET_ALL_ACTIVITY procedure on Oracle.

namespace OracleRefCursorsSM
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create the client
            SCOTTPackageACCOUNT_PKGClient accountPkgClient=
                new SCOTTPackageACCOUNT_PKGClient("OracleDBBinding_SCOTT.Package.ACCOUNT_PK
G");

            // Set credentials
            accountPkgClient.ClientCredentials.UserName.UserName = "SCOTT";
            accountPkgClient.ClientCredentials.UserName.Password = "TIGER";

            try
            {

                GET_ACTIVITYns.INOUTRECSRECORD[] strongCursor;
                GENERICns.GenRecordRow[] weakCursor;

                Console.WriteLine("Opening client");
                // Open the client
                accountPkgClient.Open();

                Console.WriteLine("Invoking ACCOUNT_PKG.GET_ACTIVITY");
                // Get ACCOUNTACTIVITY records
                // The IN REF CURSOR is set to all activity for account 100001
                // The input part of the IN OUT ref cursor calls GET_ALL_ACTIVITY
                // The weakly-typed OUT REF CURSOR parameter returns a list of activity for
account 100001
                // The strongly-typed IN OUT REF CURSOR parameter returns a list of all act
ivity
                string inRecsString = "BEGIN OPEN ? FOR SELECT * FROM ACCOUNTACTIVITY WHERE
ACCOUNT=100001; END;";
                string inoutRecsString = "BEGIN ACCOUNT_PKG.GET_ALL_ACTIVITY(?); END;";

                accountPkgClient.GET_ACTIVITY(
                    inRecsString,
                    inoutRecsString,
                    out strongCursor,
                    out weakCursor);

                // Display strong ref cursor (all activity)
                Console.WriteLine("\nList of all activity returned (strong ref cursor)");
                Console.WriteLine("Tx Id\tAccount\tAmount\tDate\t\t\tDescription");
                for (int i = 0; i < strongCursor.Length; i++)
                {
                    Console.WriteLine("{0}\t{1}\t{2:C}\t{3}\t{4}",strongCursor[i].TID,

```


Performing Operations Using RECORD Types in the WCF Service Model

Oracle RECORD types are used to represent hierarchical information in parameters passed to PL/SQL functions and procedures. The Microsoft BizTalk Adapter for Oracle Database surfaces RECORD types as complex XML types. In the WCF service model, RECORD types are deserialized to strongly-typed .NET classes. The record fields are represented as properties on the class.

The Oracle Database adapter supports the following kinds of RECORD types:

- RECORD types that are declared as TABLE%ROWTYPE parameters in stored procedures and functions.
- RECORD types that are declared as TYPE of RECORD parameters in PL/SQL packages for example, `TYPE rec_type1 IS RECORD (name varchar2(100), age number(3));`
- RECORD types that contain nested records.
- RECORD types that appear as IN, OUT, or IN OUT parameters to procedures or functions.
- RECORD types that are RETURN values of functions.

This topic shows how RECORD types are represented in the WCF service model. For information about how to call Oracle procedures and functions, see [Invoking Functions and Procedures by Using the WCF Service Model](#).

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/ACCOUNT_PKG Oracle PL/SQL PACKAGE. The following elements are used from ACCOUNT_PKG.

```
TYPE address_rec_type IS RECORD (street customer.street%TYPE, city customer.city%TYPE, state customer.state%TYPE);

FUNCTION create_account(acct IN ACCOUNT%ROWTYPE, addr IN address_rec_type) RETURN NUMBER;

TYPE acctinfo_rec_type IS RECORD (acct account%ROWTYPE, address address_rec_type);

FUNCTION get_accountinfo(aid NUMBER) RETURN acctinfo_rec_type;
```

A script to generate this package is supplied with the BizTalk Adapter Pack samples. For more information, see the script

For more information about the samples, see [Samples](#).

RECORD Types in the WCF Service Model

Oracle RECORD types are represented as complex XML types by the Oracle Database adapter. In the WCF service model, complex XML types are represented by a class, and the properties of this class represent the fields of the Oracle RECORD type. The class that represents a RECORD type parameter is generated in a namespace that is qualified by the PACKAGE (if any) and SCHEMA of the function or procedure. This namespace uniquely identifies the function or procedure of the parameter. For example, the RECORD type parameters to the CREATE_ACCOUNT procedure in the Oracle PACKAGE ACCOUNT_PKG are created in the following namespace:

`microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT`. For more information about the namespaces used in the WCF service model to represent complex types in procedures and functions, see [Invoking Functions and Procedures by Using the WCF Service Model](#).

While the namespace of a RECORD type parameter is determined by the procedure or function, the name of the class generated for the RECORD type parameter is determined by the way in which the RECORD type is declared. The following table shows how the name of the class is generated based on the two different ways of declaring the Oracle RECORD type parameter.

Oracle RECORD type	Name	Example
--------------------	------	---------

TABLE%ROWTYPE procedure or function parameter	[PARAMETER_NAME]RECORD	ACCTRECORD
TYPE of RECORD package parameter	[PACKAGE_NAME][RECORD_TYPE_NAME]RECORD	ACCOUNT_PKGACCTINFO_REC_TYPERECORD

[PARAMETER_NAME] = the name of the procedure or function parameter; for example, ACCT.

[PACKAGE_NAME] = the name of the Oracle package.

[RECORD_TYPE_NAME] = the name specified in the RECORD TYPE declaration; for example, ACCTINFO_REC_TYPE.

The following code shows the method signatures of the WCF client generated for two Oracle functions. The /SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT function takes two simple RECORD type IN parameters, and the /SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNTINFO function returns a RECORD type parameter that contains two nested RECORD types. The Oracle function declarations are included at the top of the code. The parameters of each function are qualified by a unique namespace.

```

FUNCTION create_account(acct IN ACCOUNT%ROWTYPE, addr IN address_rec_type) RETURN NUMBER;
FUNCTION get_accountinfo(aid NUMBER) RETURN acctinfo_rec_type;

public partial class SCOTTPackageACCOUNT_PKGClient : System.ServiceModel.ClientBase<SCOTTPackageACCOUNT_PKG>, SCOTTPackageACCOUNT_PKG {
    ...

    public System.Nullable<decimal> CREATE_ACCOUNT(microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT.ACCTRECORD ACCT, microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT.ACCOUNT_PKGADDRESS_REC_TYPERECORD ADDR);

    public microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCOUNT_PKGACCTINFO_REC_TYPERECORD GET_ACCOUNTINFO(System.Nullable<decimal> AID);
}

```

The following code shows the classes generated for the parameters of the CREATE_ACCOUNT function: FUNCTION create_account(acct IN ACCOUNT%ROWTYPE, addr IN address_rec_type) RETURN NUMBER;

This function has a parameter declared with a TABLE%ROWTYPE and a parameter declared with a TYPE of RECORD package type (TYPE acctinfo_rec_type IS RECORD (acct account%ROWTYPE, address address_rec_type);).

```

namespace microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT
{
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
    [System.Runtime.Serialization.DataContractAttribute()]
    public partial class ACCTRECORD : object, System.Runtime.Serialization.IExtensibleDataObject { ... }

    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
    [System.Runtime.Serialization.DataContractAttribute()]
    public partial class ACCOUNT_PKGADDRESS_REC_TYPERECORD : object, System.Runtime.Serialization.IExtensibleDataObject { ... }
}

```

Representation of a Simple Record Type

The following code shows how a simple RECORD type is represented in the WCF service model. This code shows the expanded view of the **ACCOUNTRECORD** class that represents the ACCOUNT%ROWTYPE parameter in the CREATE_ACCOUNT function. In this class, the record fields (row columns) are represented as properties.

```

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
[System.Runtime.Serialization.DataContractAttribute()]
public partial class ACCTRECORD : object, System.Runtime.Serialization.IExtensibleDataObject {

    private System.Runtime.Serialization.ExtensionDataObject extensionDataField;

    private System.Nullable<decimal> ACCTIDField;

    private string NAMEField;

    private System.Nullable<decimal> BALANCEField;

    public System.Runtime.Serialization.ExtensionDataObject ExtensionData {
        get {
            return this.extensionDataField;
        }
        set {
            this.extensionDataField = value;
        }
    }

    [System.Runtime.Serialization.DataMemberAttribute()]
    public System.Nullable<decimal> ACCTID {
        get {
            return this.ACCTIDField;
        }
        set {
            this.ACCTIDField = value;
        }
    }

    [System.Runtime.Serialization.DataMemberAttribute()]
    public string NAME {
        get {
            return this.NAMEField;
        }
        set {
            this.NAMEField = value;
        }
    }

    [System.Runtime.Serialization.DataMemberAttribute(Order=2)]
    public System.Nullable<decimal> BALANCE {
        get {
            return this.BALANCEField;
        }
        set {
            this.BALANCEField = value;
        }
    }
}

```

Representation of a Record Type that Contains Nested Records

The following code shows the representation of a RECORD type that contains nested records. This particular RECORD type is the RETURN value of the GET_ACCOUNTINFO function (FUNCTION get_accountinfo(aid NUMBER) RETURN acctinfo_rec_type;). The ACCTINFO_REC_TYPE is a package parameter declared using a TYPE of RECORD construct (TYPE acctinfo_rec_type IS RECORD (acct account%ROWTYPE, address address_rec_type;)). It contains two nested simple record types, a TABLE%ROW record and a package TYPE of RECORD record. These two simple records are declared in the same namespace as their parent record and follow the expected naming convention.

```

namespace microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO {
    using System.Runtime.Serialization;

```

```

[System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
[System.Runtime.Serialization.DataContractAttribute()]
public partial class ACCOUNT_PKGACCTINFO_REC_TYPERECORD : object, System.Runtime.Serialization.IExtensibleDataObject {

    private System.Runtime.Serialization.ExtensionDataObject extensionDataField;

    private microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCTRECORD ACCTField;

    private microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCOUNT_PKGADDRESS_REC_TYPERECORD ADDRESSField;

    public System.Runtime.Serialization.ExtensionDataObject ExtensionData {
        get {
            return this.extensionDataField;
        }
        set {
            this.extensionDataField = value;
        }
    }

    [System.Runtime.Serialization.DataMemberAttribute(IsRequired=true)]
    public microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCTRECORD ACCT {
        get {
            return this.ACCTField;
        }
        set {
            this.ACCTField = value;
        }
    }

    [System.Runtime.Serialization.DataMemberAttribute(IsRequired=true)]
    public microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCOUNT_PKGADDRESS_REC_TYPERECORD ADDRESS {
        get {
            return this.ADDRESSField;
        }
        set {
            this.ADDRESSField = value;
        }
    }
}

```

Using RECORD Types in Your Code

Using RECORD types in your code is straightforward. To invoke a procedure or function with a RECORD type parameter, you create an instance of the RECORD type or types and pass it to the appropriate method on the WCF client. When the procedure or function returns, you can read properties on any OUT or IN OUT parameters or function RETURN values that are declared as RECORD types. For more information about how to invoke procedures and functions by using the WCF service model, see [Invoking Functions and Procedures by Using the WCF Service Model](#).

◆ Important

Oracle RECORD type parameters (and function returns) are qualified by the namespace of their function or procedure (and package). This means that a RECORD type that is used in two different procedures or functions will have a different namespace for each procedure or function. You must be sure to qualify the RECORD type correctly when you use it for a specific procedure or function. For example, a package RECORD type (RECORD of TYPE declaration) that is used as an IN parameter to two different functions will be declared twice in the WCF client code with each declaration corresponding to the unique namespace generated for each function. You must be sure to use the correct namespace on the parameter that you pass to each respective function.

In the following example, the CREATE_ACCOUNT function is called with two simple record parameters. Next, the GET_ACCOUNTINFO function is called. This function returns a RECORD type that contains nested records. Selected field values from the returned RECORD are written to the console. Steps to set credentials for the Oracle database and to open the WCF client are omitted from this example.

```
// Add WCF, WCF Adapter LOB SDK, and Oracle Database adapter namespaces
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

// Include this namespace for WCF Adapter LOB SDK and Oracle Database adapter exceptions
using Microsoft.ServiceModel.Channels.Common;

...

// Create the client from configuration
using (SCOTTPackageACCOUNT_PKGClient accountPkgClient = new SCOTTPackageACCOUNT_PKGClient("
OracleDBBinding_SCOTT.Package.ACCOUNT_PKG"))
{
    ...

    decimal acctId;

    // Create an account record
    // Note: ACCTRECORD is defined in both namespaces so specify the definition
    // that corresponds to the client.

    microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT.ACCTR
ECOND acctRec =
        new microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCO
UNT.ACCTRECORD();

    // Set any value for ACCTID -- new account ID is returned by CREATE_ACCOUNT
    acctRec.ACCTID = 0;
    acctRec.NAME = "Anton Kirilov";
    acctRec.BALANCE = 9583;

    // Create address record
    microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.CREATE_ACCOUNT.ACCO
UNT_PKGADDRESS_REC_TYPERECORD addrRec = new microsoft.lobservices.oracledb._2007._03.SCOTT.P
ackage.ACCOUNT_PKG.CREATE_ACCOUNT.ACCOUNT_PKGADDRESS_REC_TYPERECORD();
    addrRec.STREET = "234 Main St";
    addrRec.CITY = "Boston";
    addrRec.STATE = "MA";

    // Create account
    try
    {
        acctId = (decimal)accountPkgClient.CREATE_ACCOUNT(acctRec, addrRec);
    }
    catch (Exception ex)
    {
        // handle exception
        ...
    }

    ...

    // Account info record
    microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.ACCO
UNT_PKGACCTINFO_REC_TYPERECORD acctInfo =
        new microsoft.lobservices.oracledb._2007._03.SCOTT.Package.ACCOUNT_PKG.GET_ACCOUNTINFO.
ACCOUNT_PKGACCTINFO_REC_TYPERECORD();

    // Get account info for the account just created
    // acctInfo is returned as a nested record type
    try
```

```
{
    acctInfo = accountPkgClient.GET_ACCOUNTINFO(acctId);
}
catch (Exception ex)
{
    // handle exception
    ...
}

// Write the account info to the console
Console.WriteLine("The account info is:");
Console.WriteLine("Name:\t\t\t{0}", acctInfo.ACCT.NAME);
Console.WriteLine("Street:\t\t\t{0}", acctInfo.ADDRESS.STREET);
Console.WriteLine("City:\t\t\t{0}", acctInfo.ADDRESS.CITY);
Console.WriteLine("State:\t\t\t{0}", acctInfo.ADDRESS.STATE);
Console.WriteLine("Account Id:\t\t{0}", acctInfo.ACCT.ACCTID);
Console.WriteLine("Account Balance:\t{0:C}", acctInfo.ACCT.BALANCE);

Console.WriteLine("\nHit <RETURN> to finish");
Console.ReadLine();
}
}
```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing a SQLEXECUTE Operation by Using the WCF Service Model

The Microsoft BizTalk Adapter for Oracle Database surfaces a standard set of operations on Oracle database artifacts. By using these operations, you can do things like call an Oracle function or procedure, or perform basic SQL data manipulation language (DML) operations on tables. However, there may be scenarios driven by your business logic that require you to perform operations that the Oracle Database adapter does not surface. For example, you may want to:

- Perform an operation on database artifacts that are not surfaced by the Oracle Database adapter; for example, get the CURVAL or NEXTVAL of an Oracle SEQUENCE.
- Perform Data Definition Language operations; for example, create a table.
- Perform operations on a database artifact that was not present at design time; for example, update records in a temporary table that is created by your business logic.
- Perform more complex DML operations on tables than the operations that the Oracle Database adapter surfaces; for example, to perform a query that includes a JOIN clause.

For these kinds of scenarios, the Oracle Database adapter surfaces the SQLEXECUTE operation. By using the SQLEXECUTE operation, you can perform a parameterized SQL statement on the Oracle database. The SQLEXECUTE operation supports an input parameter block comprised of parameter sets that enable you to execute the same SQL statement once for each set. The SQLEXECUTE operation returns the results of the SQL statement in a generic record set.

About the Examples Used in this Topic

The examples in this topic use an Oracle SEQUENCE named TID_SEQ. A script to generate this SEQUENCE is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

The WCF Client Class

The WCF service model generates a dedicated WCF client, **SQLEXECUTEClient**, for the SQLEXECUTE operation. The following code shows the **SQLEXECUTEClient** and the signature of the method that you call to invoke the SQLEXECUTE operation.

```
public partial class SQLEXECUTEClient : System.ServiceModel.ClientBase<SQLEXECUTE>, SQLEXECUTE {
    ...
    public microsoft.lobservices.oracledb._2007._03.GenRecordRow[] SQLEXECUTE(string SQLSTATEMENT, string PARAMETERSHEMA, microsoft.lobservices.oracledb._2007._03.PARAMETERDATA[] PARAMETERSET);
}
```

The SQLEXECUTE operation returns a generic record set. This record set contains the values (if any) that are returned by the statements that the SQLEXECUTE operation executes. You can pass sets of input parameters to the SQLEXECUTE operation in a collection of PARAMETERDATA objects, each of which contains a collection of input parameters represented as strings. The following code shows the definition of a PARAMETERDATA set.

```
namespace microsoft.lobservices.oracledb._2007._03 {
    using System.Runtime.Serialization;

    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
    [System.Runtime.Serialization.DataContractAttribute()]
    public partial class PARAMETERDATA : object, System.Runtime.Serialization.IExtensibleDataObject {
        ...
    }
}
```

```

private string[] PARAMETERfield;

...

[System.Runtime.Serialization.DataMemberAttribute()]
public string[] PARAMETER {
    get {
        return this.PARAMETERfield;
    }
    set {
        this.PARAMETERfield = value;
    }
}
}
}
}

```

Invoking the SQLEXECUTE Operation

To invoke the SQLEXECUTE operation by using a WCF client, perform the following steps.

1. Generate a **SQLEXECUTEclient** class for the target table or view.

◆ Important

The SQLEXECUTE operation is surfaced under the root node (/) in the **Select a category** pane in the **Add Adapter Service Reference** dialog box.

2. Create an instance of the **SQLEXECUTEclient** class, and invoke the **SQLEXECUTE** method to execute SQL statements on the Oracle database.

For more detailed information about how to create a WCF client class and invoke operations on the Oracle Database adapter, see [Overview of Using the WCF Service Model with the Adapter](#).

The following example uses the **SQLEXECUTEclient** to get the next value of an Oracle SEQUENCE, TID_SEQ, by executing the following SQL statement: `SELECT tid_seq.nextval id from DUAL`. The output is then written to the console.

```

using (SQLEXECUTEclient sqlClient = new SQLEXECUTEclient("OracleDBBinding_SQLEXECUTE"))
{
    sqlClient.ClientCredentials.UserName.UserName = "SCOTT";
    sqlClient.ClientCredentials.UserName.Password = "TIGER";
    try
    {
        sqlClient.Open();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error opening SQL client " + ex.Message);
        throw;
    }
    microsoft.lobservices.oracledb._2007._03.GenRecordRow[] sequenceRec =
        new microsoft.lobservices.oracledb._2007._03.GenRecordRow[0];

    try
    {
        sequenceRec = sqlClient.SQLEXECUTE("SELECT tid_seq.nextval id from DUAL", null, nul
1);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error executing SQL client " + ex.Message);
        throw;
    }

    if (sequenceRec.Length > 0)

```

```
    {
        Console.WriteLine("TID_SEQUENCE value is {0}", sequenceRec[0].GenRecordColumn[0].ColumnValue);
    }
    else
    {
        Console.WriteLine("Couldn't get next TID_SEQUENCE value");
    }
}
```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Receiving Polling-based Data-changed Messages by Using the WCF Service Model

You can configure the Microsoft BizTalk Adapter for Oracle Database to receive polling-based data changed messages against an Oracle table or view. To receive data-changed messages, the adapter periodically executes a SQL query against an Oracle table or view followed by an optional PL/SQL code block. The results of the SQL query are then returned by the Oracle Database adapter to your application as a strongly-typed result set in an inbound POLLINGSTMT operation. For more information about the mechanism used to configure and perform polling on an Oracle database using the Oracle Database adapter, see [Receiving Polling-based Data-changed Messages](#). We strongly recommended that you read this topic before proceeding.

To receive the POLLINGSTMT operation when you use the WCF service model, you must:

- Generate a WCF service contract (interface) for the POLLINGSTMT operation from the metadata exposed by the adapter. To do this, you use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutil.exe).
- Implement a WCF service from this interface.
- Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

The topics in this section provide information and procedures to help you perform polling on Oracle database tables and views in the WCF service model.

About the Examples Used in this Topic

The examples in this topic use the /SCOTT/ACCOUNTACTIVITY table and the /SCOTT/Package/ACCOUNT_PKG/PROCESS_ACTIVITY function. A script to generate these artifacts is supplied with the BizTalk Adapter Pack samples. For more information about the samples, see [Samples](#).

Configuring Polling in the WCF Service Model

You configure the Oracle Database adapter to perform polling on Oracle database tables and views by setting binding properties and an optional connection property (parameter). Some of these properties are mandatory, and some, to have an effect, must be set both at design-time and run-time.

- At design-time, you set connection parameters and binding properties when you connect to the Oracle Database to generate a WCF service contract.
- At runtime you set binding properties on the OracleDBBinding object that you use to create the service host. You set the connection parameter when you add a service listener to the service host.

The following list provides a brief overview of the binding properties and connection parameters used to configure polling:

- The **PollingStatement** binding property. You must set this binding property both at design-time and at run-time.
- Optional binding properties. These only have to be set at run-time.
- The **AcceptCredentialsInUri** binding property. You must set this binding property to **true** during run-time if you want to enable credentials in the connection URI. The user name and password must be present in the connection URI when you add a service endpoint to the service host.
- The **PollingId** query string parameter in the connection URI. If you want to change the namespace of the POLLINGSTMT operation, you must set this connection property both at design-time and run-time.

For a complete description of the binding properties and connection parameters used to configure polling, see [Receiving Polling-based Data-changed Messages](#).

The WCF Service Contract and Class

You use either the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutil.exe) to create a WCF service contract (interface) and supporting classes for the POLLINGSTMT operation.

When you connect to the Oracle database with either of these tools to generate a service contract for the POLLINGSTMT operation:

- You must specify the **PollingStatement** binding property. The adapter uses the SELECT statement in this binding property to generate the correct metadata for the strongly-typed result set returned by the POLLINGSTMT operation.
- You can optionally specify a PollingId parameter in the connection URI. The adapter uses this parameter to generate the namespace for the POLLINGSTMT operation.

In the following examples:

- **PollingStatement** is set to "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE".
- **PollingId** is set to "AcctActivity".

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the POLLINGSTMT operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03", ConfigurationName="POLLINGSTMT_OperationGroup")]
public interface POLLINGSTMT_OperationGroup {

    // CODEGEN: Generating message contract since the wrapper namespace (http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTAcctActivity)
    // of message POLLINGSTMT does not match the default value (http://Microsoft.LobServices.OracleDB/2007/03)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMT")]
    void POLLINGSTMT(POLLINGSTMT request);
}
```

The Message Contracts

The message contract namespace is modified by the PollingId parameter in the connection URI. The request message returns a set of strongly-typed records.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="POLLINGSTMT", WrapperNamespace="http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTAcctActivity", IsWrapped=true)]
public partial class POLLINGSTMT {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTAcctActivity", Order=0)]
    public microsoft.lobservices.oracledb._2007._03.POLLINGSTMTAcctActivity.POLLINGSTMTRECORD[] POLLINGSTMTRECORD;

    public POLLINGSTMT() {
    }

    public POLLINGSTMT(microsoft.lobservices.oracledb._2007._03.POLLINGSTMTAcctActivity.POLLINGSTMTRECORD[] POLLINGSTMTRECORD) {
        this.POLLINGSTMTRECORD = POLLINGSTMTRECORD;
    }
}
```

The Data Contract Namespace

A data contract is a formal agreement between a service and a client that abstractly describes the data to be exchanged. That is, in order to communicate, the client and the service do not have to share the same types, only the same data contracts.

In case of data change messages, the data contract namespace is also modified by the PollingId parameter (if specified) in the connection URI. The data contract is composed of a class that represents a strongly-typed record in the query result set. The details of the class definition are omitted in this example. The class contains properties that represent the columns in the result set.

In the following example, the PollingId "AcctActivity" is used.

```
namespace microsoft.lobservices.oracledb._2007._03.POLLINGSTMTAcctActivity {
    using System.Runtime.Serialization;

    [System.Diagnostics.DebuggerStepThroughAttribute()]
    [System.CodeDom.Compiler.GeneratedCodeAttribute("System.Runtime.Serialization", "3.0.0.0")]
    [System.Runtime.Serialization.DataContractAttribute(Name="POLLINGSTMTRECORD", Namespace="http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTAcctActivity")]
    public partial class POLLINGSTMTRECORD : object, System.Runtime.Serialization.IExtensibleDataObject {...}
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is OracleDBBindingService.cs. You can insert the logic to process the POLLINGSTMT operation directly into this class. If you use svcutil.exe to generate the service contract interface, you must implement this class yourself. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```
namespace OracleDBBindingNamespace {
    public class OracleDBBindingService : POLLINGSTMT_OperationGroup {
        // CODEGEN: Generating message contract since the wrapper namespace (http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMTAcctActivity)
        // of message POLLINGSTMT does not match the default value (http://Microsoft.LobServices.OracleDB/2007/03)
        public virtual void POLLINGSTMT(POLLINGSTMT request) {
            throw new System.NotImplementedException("The method or operation is not implemented.");
        }
    }
}
```

Receiving the POLLINGSTMT Operation

To receive polling data from the Oracle Database adapter

1. Use the Add Adapter Service Reference Plug-in or svcutil.exe to generate a WCF service contract (interface) and helper classes for the POLLINGSTMT operation. For more information, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#). At a minimum, you must set the **PollingStatement** binding property when you connect to the adapter. You can optionally specify a PollingId parameter in the connection URI. If you are using the Add Adapter Service Reference Plug-in, you should set all of the binding parameters necessary for your configuration. This guarantees that they are properly set in the generated configuration file.
2. Implement a WCF service from the interface and helper classes generated in step 1. The POLLINGSTMT method of this class can throw an exception to abort the polling transaction, if an error is encountered processing the data received from the POLLINGSTMT operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

- a. If you used the Add Adapter Service Reference Plug-in to generate the interface, you can implement your logic directly in the **POLLINGSTMT** method in the generated **OracleDBBindingService** class. This class can be found in OracleDBBindingService.cs. This code in this example sub-classes the **OracleDBBindingService** class.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class PollingStmtService : OracleDBBindingService
{
    public override void POLLINGSTMT(POLLINGSTMT request)
    {
        Console.WriteLine("\nNew Polling Records Received");
        Console.WriteLine("Tx Id\tAccount\tAmount\tDate\t\t\tDescription");
        for (int i = 0; i < request.POLLINGSTMTRECORD.Length; i++)
        {
            Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}", request.POLLINGSTMTRECORD[i].TID,
                                request.POLLINGSTMTRECORD[i].ACCOUNT,
                                request.POLLINGSTMTRECORD[i].AMOUNT,
                                request.POLLINGSTMTRECORD[i].TRANSDATE,
                                request.POLLINGSTMTRECORD[i].DESCRIPTION);
        }
    }
}
```

- b. If you used svcutil.exe to generate the interface, you must create a WCF service that implements the interface and implement your logic in the **POLLINGSTMT** method of this class.

3. Create an instance of the WCF service created in step 2.

```
// create service instance
PollingStmtService pollingInstance = new PollingStmtService();
```

4. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. The base connection URI cannot contain userinfoparams or a query_string.

```
// Enable service host
Uri[] baseUri = new Uri[] { new Uri("oracledb://Adapter") };
ServiceHost srvHost = new ServiceHost(pollingInstance, baseUri);
```

5. Create an **OracleDBBinding** and configure the polling operation by setting its binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the polling statement and polling interval. In this example, you specify the credentials as part of the URI so you must also set the **AcceptCredentialsInUri** to **true**.

```
// Create and configure a binding for the service endpoint. NOTE: binding
// parameters are set here for clarity, but these are already set in the
// the generated configuration file
OracleDBBinding binding = new OracleDBBinding();

// The credentials are included in the connection URI, so set this property to true
binding.AcceptCredentialsInUri = true;

// Same as statement specified in Configure Adapter dialog box
binding.PollingStatement = "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE";
binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";
```

```
// Be sure to set the interval long enough to complete processing before
// the next poll
binding.PollingInterval = 15;
// Polling is transactional; be sure to set an adequate isolation level
// for your environment
binding.TransactionIsolationLevel = TransactionIsolationLevel.ReadCommitted;
```

6. Add a service endpoint to the service host. To do this:

- Use the binding created in step 5.
- Specify a connection URI that contains credentials and, if needed, a PollingId.
- Specify the contract as "POLLINGSTMT_OperationGroup".

```
// Add service endpoint: be sure to specify POLLINGSTMT_OperationGroup as the contract
Uri serviceUri = new Uri("oracledb://User=SCOTT;Password=TIGER@Adapter?PollingId=AcctA
ctivity");
srvHost.AddServiceEndpoint("POLLINGSTMT_OperationGroup", binding, serviceUri);
```

7. To receive polling data, open the service host. The adapter will return data whenever the query returns a result set.

```
// Open the service host to begin polling
srvHost.Open();
```

8. To terminate polling, close the service host.

◆ Important

The adapter will continue to poll until the service host is closed.

```
srvHost.Close();
```

Example

The following example shows a polling query that executes against the /SCOTT/ACCOUNTACTIVITY table. The post-poll statement invokes an Oracle function that moves the processed records to another table /SCOTT/ACCOUNTHISTORY. The namespace of the POLLINGSTMT operation is modified by setting the PollingId parameter to "AccountActivity" in the connection URI. In this example, the WCF service for the POLLINGSTMT operation is created by sub-classing the generated **OracleDBBindingService** class; however, you can implement your logic directly in the generated class.

```
using System;
using System.Collections.Generic;
using System.Text;

// Add these three references to use the Oracle adapter
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

using microsoft.lobservices.oracledb._2007._03.POLLINGSTMTAcctActivity;
using OracleDBBindingNamespace;

namespace OraclePollingSM
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class PollingStmtService : OracleDBBindingService
    {
```

```

public override void POLLINGSTMT(POLLINGSTMT request)
{
    Console.WriteLine("\nNew Polling Records Received");
    Console.WriteLine("Tx Id\tAccount\tAmount\tDate\t\t\tDescription");
    for (int i = 0; i < request.POLLINGSTMTRECORD.Length; i++)
    {
        Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}", request.POLLINGSTMTRECORD[i].T
ID,
                                request.POLLINGSTMTRECORD[i].ACCOUNT,
                                request.POLLINGSTMTRECORD[i].AMOUNT,
                                request.POLLINGSTMTRECORD[i].TRANSDATE,
                                request.POLLINGSTMTRECORD[i].DESCRIPTION);

    }
    Console.WriteLine("\nHit <RETURN> to stop polling");
}
}

class Program
{
    static void Main(string[] args)
    {
        ServiceHost srvHost = null;

        // This URI is used to specify the address for the ServiceEndpoint
        // It must contain credentials and the PollingId (if any) that was used to gene
rate
        // the WCF service callback interface
        Uri serviceUri = new Uri("OracleDb://User=SCOTT;Password=TIGER@Adapter?PollingI
d=AcctActivity");

        // This URI is used to initialize the ServiceHost. It cannot contain
        // userinfoparms (credentials) or a query_string (PollingId); otherwise,
        // an exception is thrown when the ServiceHost is initialized.
        Uri[] baseUri = new Uri[] { new Uri("OracleDb://Adapter") };

        Console.WriteLine("Sample started, initializing service host -- please wait");

        // create an instanc of the WCF service callback class
        PollingStmtService pollingInstance = new PollingStmtService();

        try
        {
            // Create a ServiceHost with the service callback instance and a base URI (
address)
            srvHost = new ServiceHost(pollingInstance, baseUri);

            // Create and configure a binding for the service endpoint. Note: binding
            // parameters are set here for clarity but these are already set in the
            // generated configuration file
            //
            // The following properties are set
            //   AcceptCredentialsInUri (true) to enable credentials in the connection
URI for AddServiceEndpoint
            //   PollingStatement
            //   PostPollStatement calls PROCESS_ACTIVITY on Oracle. This procedure mo
ves the queried records to
            //           the ACCOUNTHISTORY table
            //   PollingInterval (15 seconds)
            //   TransactionIsolationLevel

            OracleDBBinding binding = new OracleDBBinding();

            // The Credentials are included in the Connection Uri so set this property
true
            binding.AcceptCredentialsInUri = true;

            // Same as statement specified in Configure Adapter dialog box

```

```

binding.InboundOperationType = InboundOperation.Polling;
binding.PollingStatement = "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE";
binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";

// Be sure to set the interval long enough to complete processing before
// the next poll
binding.PollingInterval = 15;

// Polling is transactional, be sure to set an adequate isolation level
// for your environment
binding.TransactionIsolationLevel = TransactionIsolationLevel.ReadCommitted
;

// Add service endpoint: be sure to specify POLLINGSTMT_OperationGroup as t
he contract
srvHost.AddServiceEndpoint("POLLINGSTMT_OperationGroup", binding, serviceUr
i);

Console.WriteLine("Opening the service host");
// Open the service host to begin polling
srvHost.Open();

// Wait to receive request
Console.WriteLine("\nPolling started. Returned records will be written to t
he console.");
Console.WriteLine("Hit <RETURN> to stop polling");
Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Exception :" + e.Message);
    Console.ReadLine();

    /* If there is an Oracle Error it will be specified in the inner exception
*/
    if (e.InnerException != null)
    {
        Console.WriteLine("InnerException: " + e.InnerException.Message);
        Console.ReadLine();
    }
}
finally
{
    // IMPORTANT: you must close the ServiceHost to stop polling
    if (srvHost.State == CommunicationState.Opened)
        srvHost.Close();
    else
        srvHost.Abort();
}
}
}
}

```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Receiving Database Change Notifications Using the WCF Service Model

This topic demonstrates how to configure the Oracle Database adapter to receive query notification messages from an Oracle database. To demonstrate notifications, consider a table, ACCOUNTACTIVITY, with a "Processed" column. When a new record is inserted to this table, the value of the Status column is set to 'n'. You can configure the adapter to receive notifications by registering for notifications using a SQL statement that retrieves all records that have "Processed" column as 'n'. You can do so by specifying the SQL statement for the **NotificationStatement** binding property. Once the adapter client receives the notification, it can contain the logic to do any subsequent tasks on the Oracle database. In this example, for the sake of simplicity, the adapter client lists all the records in the table that have the "Processed" column as 'n'.

Configuring Notifications with the Oracle Database Adapter Binding Properties

The table below summarizes the Oracle Database adapter binding properties that you use to configure receiving notifications from Oracle database. You must specify these binding properties while running the .NET application to receive notifications.

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the SELECT statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified SELECT statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For a complete description of how to use the Oracle Database adapter to receive notifications from Oracle database, read further.

Configuring Notifications Using the WCF Service Model

To receive the notifications using the WCF service model, you must:

- Generate a WCF service contract (interface) for the **Notification** operation from the metadata exposed by the adapter. To do this, you could use the Add Adapter Service Reference Plug-in.
- Generate a WCF client for the **Select** operation on the ACCOUNTACTIVITY table. To do this, you could use the Add Adapter Service Reference Plug-in.
- Implement a WCF service from this interface.
- Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **Notification** operation. For more information about generating a WCF service contract, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **Notification** operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
```

```
[System.ServiceModel.ServiceContractAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03", ConfigurationName="Notification_OperationGroup")]
public interface Notification_OperationGroup {

    // CODEGEN: Generating message contract since the wrapper namespace (http://Microsoft.LobServices.OracleDB/2007/03/Notification/) of message Notification
    // does not match the default value (http://Microsoft.LobServices.OracleDB/2007/03)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="http://Microsoft.LobServices.OracleDB/2007/03/Notification")]
    void Notification(Notification request);
}
```

The Message Contracts

Following is the message contract for the Notification operation.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="Notification", WrapperNamespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", IsWrapped=true)]
public partial class Notification {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", Order=0)]
    public microsoft.lobservices.oracledb._2007._03.Notification.NotificationDetails[] Details;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", Order=1)]
    public string Info;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", Order=2)]
    public string[] ResourceNames;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", Order=3)]
    public string Source;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://Microsoft.LobServices.OracleDB/2007/03/Notification/", Order=4)]
    public string Type;

    public Notification() {
    }

    public Notification(microsoft.lobservices.oracledb._2007._03.Notification.NotificationDetails[] Details, string Info, string[] ResourceNames, string Source, string Type) {
        this.Details = Details;
        this.Info = Info;
        this.ResourceNames = ResourceNames;
        this.Source = Source;
        this.Type = Type;
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is OracleDBBindingService.cs. You can insert the logic to process the **Notification** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```
namespace OracleDBBindingNamespace {

    public class OracleDBBindingService : Notification_OperationGroup {
```



```

        Console.WriteLine("The application will now select the last inserted record");

        microsoft.lobservices.oracledb._2007._03.SCOTT.Table.ACCOUNTACTIVITY.ACCOUNTACTIVITYRECORDSELECT[] selectRecords;

        try
        {
            selectRecords = client.Select("*", "WHERE PROCESSED = 'n'");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Exception: " + ex.Message);
            throw;
        }

        Console.WriteLine("The details of the newly added records are:");
        Console.WriteLine("*****");
        for (int i = 0; i < selectRecords.Length; i++)
        {
            Console.WriteLine("Transaction ID   : " + selectRecords[i].TID);
            Console.WriteLine("Account ID      : " + selectRecords[i].ACCOUNT);
            Console.WriteLine("Processed Status : " + selectRecords[i].PROCESSED);
            Console.WriteLine();
        }
        Console.WriteLine("*****");
    }
}
}

```

2. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **Notification** operation.

For more information, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

3. Implement a WCF service from the interface and helper classes generated in step 2. The **Notification** method of this class can throw an exception to abort the operation, if an error is encountered processing the data received from the **Notification** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **Notification** method, you can implement your application logic directly. This class can be found in OracleDBBindingService.cs. This code in this example sub-classes the **OracleDBBindingService** class. In this code, the notification message received is written to the console. Additionally, the **TableOp** method within the **TableOperation** class is invoked to perform the Select operation.

```

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class NotificationService : OracleDBBindingNamespace.OracleDBBindingService
{
    public override void Notification(Notification request)
    {
        Console.WriteLine("\nNew Notification Received");
        Console.WriteLine("*****");
    }
}

```

```

        Console.WriteLine(request.Info);
        Console.WriteLine(request.Source);
        Console.WriteLine(request.Type);
        Console.WriteLine("*****");

        TableOperation Ops = new TableOperation();
        Ops.TableOp();

    }
}

```

4. You must implement the following class to pass credentials for the Oracle database. In the latter part of the application, you will instantiate this class to pass on the credentials.

```

class NotificationCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new NotificationCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

```

5. Create an **OracleDBBinding** and configure the adapter to receive query notifications by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType** and **NotificationStatement** binding properties.

```

OracleDBBinding binding = new OracleDBBinding();
binding.InboundOperationType = InboundOperation.Notification;
binding.NotificationStatement = "SELECT TID,ACCOUNT,PROCESSED FROM APPS.ACCOUNTACTIVITY WHERE PROCESSED = 'n'";
binding.NotifyOnListenerStart = true;
binding.NotificationPort = 10;

```

◆ Important

The value for the **NotificationPort** binding property must be set to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkId=196959>.

◆ Important

If you do not set the **NotificationPort** binding property, the adapter will assume the default value of -1 for this binding property. In such a case, you will have to completely disable Windows Firewall to receive notification messages.

6. Specify Oracle database credentials by instantiating the **NotificationCredentials** class you created in Step 4.

```
NotificationCredentials credentials = new NotificationCredentials();
credentials.UserName.UserName = "SCOTT";
credentials.UserName.Password = "TIGER";
```

7. Create an instance of the WCF service created in step 3.

```
// create service instance
NotificationService service = new NotificationService();
```

8. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. You must also specify the credentials here.

```
// Enable service host
Uri[] baseUri = new Uri[] { new Uri("oracledb://adapter") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);
```

9. Add a service endpoint to the service host. To do this:

- Use the binding created in step 5.
- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "Notification_OperationGroup".

```
// Add service endpoint: be sure to specify Notification_OperationGroup as the contract
Uri ConnectionUri = new Uri("oracledb://adapter");
serviceHost.AddServiceEndpoint("Notification_OperationGroup", binding, ConnectionUri);
```

10. To receive notification message, open the service host.

```
// Open the service host to begin receiving notifications
serviceHost.Open();
```

11. To stop receiving notifications, close the service host.

```
serviceHost.Close();
```

Example

The following example shows a .NET application to receive notification messages for the ACCOUNTACTIVITY table.

Note

The following code snippet instantiates a **TableOperation.cs** class and invokes the **TableOp** method. The class and the method are described in Step 1.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Adapters.OracleDB;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace Notification_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class NotificationService : OracleDBBindingNamespace.OracleDBBindingService
    {
        public override void Notification(Notification request)
        {
            Console.WriteLine("\nNew Notification Received");
            Console.WriteLine("*****");
            Console.WriteLine(request.Info);
            Console.WriteLine(request.Source);
            Console.WriteLine(request.Type);
            Console.WriteLine("*****");

            TableOperation Ops = new TableOperation();
            Ops.TableOp();

        }
    }

    class NotificationCredentials : ClientCredentials, IServiceBehavior
    {
        public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
dingParameters)
        {
            bindingParameters.Add(this);
        }

        public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
tBase serviceHostBase)
        { }

        public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
HostBase)
        { }

        protected override ClientCredentials CloneCore()
        {
            ClientCredentials clone = new NotificationCredentials();
            clone.UserName.UserName = this.UserName.UserName;
            clone.UserName.Password = this.UserName.Password;
            return clone;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            ServiceHost serviceHost = null;
            try
            {
                Console.WriteLine("Sample started...");
                Console.WriteLine("Press any key to start receiving notifications...");
            }
        }
    }
}

```


Developing Applications by Using the WCF Channel Model

You can use the Windows Communication Foundation (WCF) channel model to consume the Microsoft BizTalk Adapter for Oracle Database by sending XML messages directly over a channel instance created with the Oracle DB Binding.

One advantage of using the WCF channel model over using the strongly-typed classes and methods that the WCF service model exposes is that the channel model provides more fine-grained control over the operations that you perform on the Oracle database. Why? In the WCF channel model you directly control the contents of the messages that you send over the channel.

In certain scenarios, this extra level of control can be beneficial. For example, when you use the WCF channel model to perform an Update operation on a table, you can selectively update columns in the target rows by omitting columns from the update template that you pass in the message. The update method exposed by a WCF client uses a strongly-typed record parameter for the template that includes every column in the table schema. If a column has "nillable=false" in the WSDL, it must be updated using the WCF service model.

Another key advantage that the WCF channel model provides over the WCF service model is more comprehensive support for end-to-end streaming of Oracle large object (LOB) data types. By using the WCF channel model you can perform end-to-end streaming:

- To update an LOB column in a table or view using the UpdateLOB operation.
- On OUT and IN OUT parameters containing LOB data that are returned by procedures and functions.
- On LOB data that is contained in the result of a SQLEXECUTE operation.
- On LOB data columns that are returned in the POLLINGSTMT operation.
- On LOB data columns that are returned by a Select operation on a table or view.

This is because in the WCF channel model you directly control how you provide the message body on outgoing messages and how you process the message body on incoming messages.

In contrast, the WCF service model only provides:

- End-to-end streaming for LOB data on one operation, the ReadLOB operation.
- No capability to update LOB data on the Oracle database in a streamed fashion.

The sections in this topic explain how to perform operations on the Oracle Database adapter by using the WCF channel model.

In This Section

- [Overview of Using the WCF Channel Model with the Adapter](#)
- [Creating a Channel](#)
- [Invoking Operations on the Oracle Database by Using the WCF Channel Model](#)
- [Performing a SQLEXECUTE Operation by Using the WCF Channel Model](#)
- [Performing an Insert Operation by Using the WCF Channel Model](#)
- [Invoking a Function by Using the WCF Channel Model](#)
- [Receiving Polling-based Data-changed Messages by Using the WCF Channel Model](#)
- [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#)

Overview of Using the WCF Channel Model with the Adapter

To invoke operations on the Microsoft BizTalk Adapter for Oracle Database, your code acts as a WCF client and sends outbound operations to the adapter. In the WCF channel model, your code invokes operations on the adapter by sending a request message over a channel.

To invoke inbound operations, such as receiving polling-based data-changed messages using the POLLINGSTMT operation provided by the adapter, your code acts as a WCF service and receives the inbound operation from the adapter. In other words, your code receives a request message from the adapter over a channel.

The topics in this section provide an overview of using the Oracle Database adapter with the WCF channel model.

WCF Channel Model Overview

Clients and services communicate by exchanging SOAP messages. The WCF channel model is a low-level abstraction of this message exchange. It provides interfaces and types that enable you to send and receive messages by using a layered protocol stack called a channel stack. Each layer of the stack is composed of a channel, and each channel is created from a WCF binding. At the lowest layer is the transport channel. The transport channel implements the underlying transport mechanism between a service and a client and presents each message to the higher layers (and ultimately the consuming application) as a **System.ServiceModel.Message**. The WCF **Message** class is an abstraction of a SOAP message. WCF provides several channel interfaces, called channel shapes, that model the basic SOAP message exchange patterns, such as request-reply or one-way. A WCF transport binding provides an implementation of one or more channel shapes that higher layers can use to send and receive messages. For more information about the WCF channel model, see "Channel Model Overview" at <http://go.microsoft.com/fwlink/?LinkId=82614>.

The Oracle Database adapter is a WCF custom transport binding that exposes an Oracle database as a WCF service.

Supported Channel Shapes for the Oracle Database Adapter

The adapter implements the following WCF channel shapes:

- **IRequestChannel (System.ServiceModel.Channels.IRequestChannel)**. The **IRequestChannel** interface implements the client side of a request-reply message exchange. You can use an **IRequestChannel** to perform operations for which you want to consume a response, for example to perform a SELECT query on an Oracle table.
- **IOutputChannel (System.ServiceModel.Channels.IOutputChannel)**. This shape implements the client side of a one-way message exchange. You can use an **IOutputChannel** to invoke an operation for which you do not need to consume a response, for example to call an Oracle procedure that has no OUT parameters.

◆ Important

All underlying calls by the adapter to the Oracle client are synchronous. This includes calls to the Oracle client that are the result of operations invoked over an **IOutputChannel**. When you use an **IOutputChannel**, the adapter discards the response received from the Oracle client.

- **IInputChannel (System.ServiceModel.Channels.IInputChannel)**. This shape implements the service side of a one-way message exchange. You use an **IInputChannel** to receive messages for inbound operations from the adapter.

Like any WCF binding, the Oracle Database adapter uses a factory pattern to provide channels to application code. You use a **Microsoft.Adapters.OracleDBBinding** object to create instances of:

- **System.ServiceModel.ChannelFactory<IRequestChannel>** to provide **IRequestChannel** channels you can use to invoke request-response operations on the adapter.
- **System.ServiceModel.ChannelFactory<IOutputChannel>** to provide **IOutputChannel** channels you can use to invoke one-way operations on the adapter.
- **System.ServiceModel.ChannelListener<IInputChannel>** to provide **IInputChannel** channels you can use to receive inbound messages (e.g. POLLINGSTMT operation) from the adapter.

Creating Messages for the Oracle Database Adapter in the WCF Channel Model

In WCF the **System.ServiceModel.Channels.Message** class provides an in memory representation of a SOAP message. You create a **Message** instance by invoking the static **Message.Create** method.

There are two important parts to the SOAP message that you must specify when you create a **Message** instance to send to the Oracle Database adapter.

- The message action is a string that is part of the SOAP message header. The message action identifies the operation that should be invoked on the Oracle database. The following shows the message action specified to invoke the Select operation on the /SCOTT/EMP table: `http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select`.
- The message body contains the parameter data for the operation. The message body is composed of well-formed XML that corresponds to the message schema expected by the Oracle Database adapter for the requested operation. The following message body specifies a Select operation on the SCOTT.EMP table (SELECT * FROM EMP).

```
<?xml version="1.0" encoding="utf-8" ?>
<Select xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP">
  <COLUMN_NAMES>*</COLUMN_NAMES>
</Select>
```

For information about the Oracle Database adapter message schemas and message actions for operations, see [Messages and Message Schemas for BizTalk Adapter for Oracle Database](#).

This **Create** method is overloaded and offers many different options for providing the message body. The following code shows how to create a **Message** instance by using an **XmlReader** to supply the message body. In this code, the message body is read from a file.

```
XmlReader readerIn = XmlReader.Create("SelectAllActivity.xml");
Message messageIn = Message.CreateMessage(MessageVersion.Default,
    "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select",
    readerIn);
```

◆ Important

You must provide a message action in your **Message** instance. This is typically done when the **Message** instance is created.

Streaming Support for LOB Data Types in the WCF Channel Model

End-to-end streaming of LOB data types is supported for some operations surfaced by the adapter. For these operations, how you create and consume the messages that you send and receive over the channel determines whether streaming is supported on the LOB data.

For more information about how the Oracle Database adapter supports streaming on LOB data, see [Streaming and the Oracle Database Adapter](#).

For more information about implementing node-value streaming in your code to support end-to-end streaming of LOB data, see [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#).

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Creating a Channel

In the WCF channel model, you invoke operations on the Oracle database and receive the results of a polling query by exchanging SOAP messages with the Microsoft BizTalk Adapter for Oracle Database over a WCF channel.

- You invoke operations (outbound operations) by using either an **IRequestChannel** or an **IOutputChannel** to send messages to the adapter.
- You receive polling-based data-changed messages by receiving POLLINGSTMT messages over an **IInputChannel**.

The topics in this section provide information about how to create and configure channel shapes that are used for inbound and outbound operations.

Creating Outbound (Client) Channels

You can use either an **IRequestChannel** or an **IOutputChannel** to invoke operations on the Oracle database. In either case, you first create a **System.ServiceModel.ChannelFactory** using the appropriate interface. You then use the factory to create the channel. After you have created the channel you can use it to invoke operations on the adapter.

To create and open an outbound channel

1. Create and initialize an instance of **ChannelFactory** for the desired channel shape by using an endpoint and a binding. The endpoint specifies an Oracle connection URI and the binding is an instance of **OracleDBBinding**.
2. Provide Oracle credentials for the channel factory by using the **Credentials** property.
3. Open the channel factory.
4. Get an instance of the channel by invoking the **CreateChannel** method on the channel factory.
5. Open the channel.

You can specify the binding and endpoint address in your code or from configuration.

Specifying the Binding and Endpoint Address in Code

The following code example shows how to create an **IRequestChannel** by specifying the binding and endpoint address in code. The code to create an **IOutputChannel** is the same except that you must specify an **IOutputChannel** interface for the **ChannelFactory** and channel type.

```
// Create binding -- set binding properties before you open the factory.
OracleDBBinding odbBinding = new OracleDBBinding();

// Create address.
EndpointAddress odbAddress = new EndpointAddress("oracledb://ADAPTER/");

// Create channel factory from binding and address.
ChannelFactory<IRequestChannel> factory =
    new ChannelFactory<IRequestChannel>(odbBinding, odbAddress);

// Specify credentials.
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";

// Open factory
factory.Open();

// Get channel and open it
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

Specifying the Binding and Endpoint Address in Configuration

The following code example shows how to create a channel factory from a client endpoint specified in configuration.

```

// Create channel factory from configuration.
ChannelFactory<IRequestChannel> factory =
new ChannelFactory<IRequestChannel>("MyRequestChannel");

// Specify credentials.
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";

// Open the factory.
factory.Open();

// Get a channel and open it.
IRequestChannel channel = factory.CreateChannel();
channel.Open();

```

The Configuration Settings

The following code shows the configuration settings used for the preceding example. The contract for the client endpoint must be "System.ServiceModel.Channels.IRequestChannel" or "System.ServiceModel.Channels.IRequestChannel" depending on the kind of channel shape that you want to create.

```

<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <oracleDBBinding>
        <binding name="OracleDBBinding" closeTimeout="00:01:00" openTimeout="00:01:
00"
          receiveTimeout="00:10:00" sendTimeout="00:01:00" metadataPooling="true"
          statementCachePurge="false" statementCacheSize="10" pollingInterval="50
0"
          useOracleConnectionPool="true" minPoolSize="1" maxPoolSize="100"
          incrPoolSize="5" decrPoolSize="1" connectionLifetime="0" acceptCredenti
alsInUri="false"
          useAmbientTransaction="true" polledDataAvailableStatement="SELECT 1 FRO
M DUAL"
          pollWhileDataFound="false" notifyOnListenerStart="true" notificationPor
t="-1"
          inboundOperationType="Polling" dataFetchSize="65536" longDatatypeColumn
Size="0"
          skipNilNodes="true" maxOutputAssociativeArrayElements="32"
          enableSafeTyping="false" insertBatchSize="1" useSchemaInNamespace="true
"
          enableBizTalkCompatibilityMode="false" enablePerformanceCounters="false
" />
      </oracleDBBinding>
    </bindings>
    <client>
      <endpoint address="oracledb://adapter/" binding="oracleDBBinding"
        bindingConfiguration="OracleDBBinding" contract="System.ServiceModel.Channe
ls.IRequestChannel"
        name="MyRequestChannel" />
    </client>
  </system.serviceModel>
</configuration>

```

Creating Inbound (Service) Channels

You configure the Oracle Database adapter to poll the Oracle database tables and views by setting binding properties on an instance of **OracleDBBinding**. You then use this binding to build a channel listener from which you can get an **IInputChannel** channel to receive message for inbound operations from the adapter.

To create and open an IInputChannel to receive messages for inbound operations

1. Create an instance of **OracleDBBinding**.
2. Set the binding properties required for the inbound operation. For example, for the POLLINGSTMT operation, at a

minimum you must set the **InboundOperationType**, **PollingStatement**, and **PollingInterval** binding properties to configure the Oracle Database adapter to poll the Oracle database.

3. Create a binding parameter collection using the **BindingParameterCollection** class and set the credentials.
4. Create a channel listener by invoking **BuildChannelListener<IInputChannel>** method on the **OracleDBBinding**. You specify the Oracle connection URI as one of the parameters to this method. For more information about the Oracle connection URI, see [The Oracle Database Connection URI](#).
5. Open the listener.
6. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on listener.
7. Open the channel.

The following code shows how to create a channel listener and get an **IInputChannel** to inbound messages from the adapter using the POLLINGSTMT operation.

Note

The Oracle Database adapter only supports one-way receive. So, you must use **IInputChannel** to receive messages for inbound operations from Oracle database.

```
// Create a binding: specify the InboundOperationType, PollingInterval (in seconds), the Po
llingStatement, and
// the PostPollStatement.
OracleDBBinding binding = new OracleDBBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PollingInterval = 30;
binding.PollingStatement = "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE";
binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";

// Create a binding parameter collection and set the credentials
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.UserName = "SCOTT";
credentials.UserName.Password = "TIGER";

BindingParameterCollection bindingParams = new BindingParameterCollection();
bindingParams.Add(credentials);

// Get a listener from the binding and open it.
Uri connectionUri = new Uri("oracleDB://ADAPTER");
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>(conn
ectionUri, bindingParams);
listener.Open();

// Get a channel from the listener and open it.
channel = listener.AcceptChannel();
channel.Open();
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Invoking Operations on the Oracle Database by Using the WCF Channel Model

You can invoke operations on the Oracle Database adapter by using an **IRequestChannel** or **IOutputChannel** shape to send messages to the adapter. The basic pattern is to create a channel factory for the required channel shape by using a binding (**OracleDBBinding**) and an endpoint created from a connection URI. You then create a **Message** instance that represents a SOAP message that conforms to the message schema for your target operation. You can then send this **Message** to the Oracle Database adapter by using a channel created from the channel factory. If you are using an **IRequestChannel**, you receive a response. If there is a problem executing the operation on the Oracle database, the Oracle Database adapter throws a **Microsoft.ServiceModel.Channels.Common.TargetSystemException**.

For an overview of how to send operations using an **IRequestChannel** in WCF, see "Client Channel-Level Programming" at <http://go.microsoft.com/fwlink/?LinkId=106081>.

The sections in this topic provide information to help you invoke operations on the Oracle Database adapter using the WCF channel model.

Creating and Consuming Messages for Outbound Operations

To invoke an operation on the Oracle Database adapter, you send the request message for the target operation using either an **IRequestChannel** or an **IOutputChannel**. If you use an **IRequestChannel** the adapter returns the results of the operation in the response message.

For more detailed information about the request and response message schemas and the message actions for each operation, see [Messages and Message Schemas for BizTalk Adapter for Oracle Database](#).

How you create the request message and consume the response message determines whether node streaming or node-value streaming is performed by the adapter. This in turn determines whether end-to-end streaming of LOB data is performed for supported operations.

Creating the request message

You can create the request message in one of two ways:

- To create a message that can be used for node-value streaming you must pass the message body in an **XmlBodyWriter** that implements node-value streaming.
- To create a message that can be used for node streaming you can pass the message body in an **XmlReader**.

You typically use node-value streaming to support end-to-end streaming of Oracle LOB data in the request message. The only operation that supports this feature is UpdateLOB.

Consuming the response message

You can consume the response message in one of two ways:

- To consume the message using node-value streaming you must call the **WriteBodyContents** method on the response message and pass it an **XmlDictionaryWriter** that implements node-value streaming.
- To consume the message using node streaming you can call **GetReaderAtBodyContents** on the response message to get an **XmlReader**.

You typically use node-value streaming to support end-to-end streaming of Oracle LOB data in the response message. There are many operations that support this feature.

LOB Data and Message Streaming Support

For more information about how the Oracle Database adapter supports streaming on LOB data, see [Streaming and the Oracle Database Adapter](#).

For more information about implementing node-value streaming in your code to support end-to-end streaming of LOB data, see [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#).

Transaction Support on Outbound Operations in the WCF Channel Model.

The adapter executes each operation you invoke inside a dedicated transaction on the Oracle database. You can control the isolation level of these transactions by setting the **TransactionIsolationLevel** binding property.

About the Examples Used in this Topic

The example in this topic uses the SCOTT.ACCOUNTACTIVITY table. A script to generate these artifacts is supplied with the SDK samples. For more information about the SDK samples, see [Samples](#).

How Do I Invoke an Operation by Using a Channel?

To invoke an operation by using an **IRequestChannel**, perform the following steps.

How to invoke an operation by using an instance of IRequestChannel

1. Build a channel factory (**ChannelFactory<IRequestChannel>**). To do this, you must specify a binding (**OracleDBBinding**) and an endpoint address. You can specify the binding and endpoint address either imperatively in your code or declaratively in configuration. For more information about how to specify the binding and endpoint address in configuration, see [Creating a Channel](#).

```
// Create a binding
OracleDBBinding binding = new OracleDBBinding();
// Create an endpoint address by using the connection URI
EndpointAddress address = new EndpointAddress("oracledb://ADAPTER");
// Create the channel factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding,
address);
```

2. Set the user name password credentials for the channel factory by using the **ClientCredentials** property.

```
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";
```

3. Open the channel factory.

```
factory.Open();
```

4. Get a channel from the factory and open it.

```
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

5. Create a **Message** instance for the target operation. Be sure that the message action for the target operation is specified. In this example, the message body is passed by creating an **XmlReader** over a file. The target operation is a Select operation on the SCOTT/EMP table.

```
XmlReader readerIn = XmlReader.Create("SelectAllActivity.xml");
Message messageIn = Message.CreateMessage(MessageVersion.Default,
    "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY/Select"
    ,
    readerIn);
```

6. Invoke the **Request** method on the channel to send the message to the Oracle Database adapter and receive the reply. If the Oracle database encounters an exception, the adapter throws a **TargetSystemException**. (Other exceptions are possible for non Oracle exceptions.) You can get a description of the Oracle error from the **InnerException.Message** property of the **TargetSystemException**.

```
try
{
```

```

        Message messageOut = channel.Request(messageIn);
    }
    catch (Exception ex)
    {
        // handle exception
    }

```

7. Process the response. In this example, **GetReaderAtBodyContents** is called on the response message to get the message body.

```

XmlReader readerOut = messageOut.GetReaderAtBodyContents();

```

8. When you are done processing the response message, close the reader and the message.

```

readerOut.Close();
messageOut.Close();

```

9. When you are done using the channel and the channel factory, close them. Closing the factory will close all channels that were created with it.

```

channel.Close()
factory.Close();

```

You follow the same steps to send a message using the **IOutputChannel** shape except:

- You create a **ChannelFactory<IOutputChannel>** in step 1.
- You call the **Send** method on the channel in step 6. `channel.Send(messageIn);`.
- There is no response message returned for an **IOutputChannel**.

Example

The following example shows how to invoke a Select operation by using an **IRequestChannel** channel. The Select response message is consumed by using an **XmlReader** and the number of records returned is written to the console.

```

using System;
using System.Collections.Generic;
using System.Text;

using System.ServiceModel;
using System.ServiceModel.Channels;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;
using System.Xml;
using System.IO;
using System.Runtime.Serialization;

namespace RequestChanneSample
{
    class Program
    {
        static void Main(string[] args)
        {
            // The Select operation request message
            const string selectRequestString =
                "<Select xmlns=\"http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY\">" +

```

```

        "<COLUMN_NAMES>*</COLUMN_NAMES>" +
        "<FILTER>ACCOUNT = 100002</FILTER>" +
        "</Select>";
try
{
    // Create binding -- specify binding properties before you open the factory
    OracleDBBinding odbBinding = new OracleDBBinding();

    // Create address.
    EndpointAddress odbAddress = new EndpointAddress("oracledb://ADAPTER/");

    // Create channel factory from binding and address.
    ChannelFactory<IRequestChannel> factory =
        new ChannelFactory<IRequestChannel>(odbBinding, odbAddress);

    // Specify credentials
    factory.Credentials.UserName.UserName = "SCOTT";
    factory.Credentials.UserName.Password = "TIGER";

    // Open the factory.
    factory.Open();

    // Get a channel.
    IRequestChannel channel = factory.CreateChannel();

    // Open the channel.
    channel.Open();

    // Create the request message from the string
    StringReader strReader = new StringReader(selectRequestString);
    XmlReader readerIn = XmlReader.Create(strReader);

    Message requestMessage = Message.CreateMessage(MessageVersion.Default,
        "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIV
ITY/Select",
        readerIn);

    Send the message and get a response
    Message responseMessage = channel.Request(requestMessage);

    // Get an XmlReader from the message
    XmlReader readerOut = (XmlReader) responseMessage.GetReaderAtBodyContents()
;

    // Count the number of records returned and write to the console.
    readerOut.MoveToContent();
    int numberOfRecordsReturned = 0;
    while (readerOut.Read())
    {
        if (readerOut.NodeType == XmlNodeType.Element && readerOut.Name == "ACC
OUNTACTIVITYRECORDSELECT")
            numberOfRecordsReturned++;
    }

    Console.WriteLine("{0} records returned.", numberOfRecordsReturned);

    // Close the output reader and message
    readerOut.Close();
    responseMessage.Close();

    //Close channel
    channel.Close();

    //Close the factory
    factory.Close();
}
catch (Exception ex)

```

```
}  
  }  
    }  
      {  
        Console.WriteLine(ex.Message);  
      }  
    }  
  }  
}
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Performing a SQLEXECUTE Operation by Using the WCF Channel Model

This section shows how to perform a SQLEXECUTE operation on an Oracle database over a channel. You must specify both a message and a message action on the SOAP message. For more information about the SQLEXECUTE operation, see [Performing a SQLEXECUTE Operation by Using the WCF Service Model](#).

The SQLEXECUTE Message

The following XML shows a SQLEXECUTE message that returns the next value of an Oracle SEQUENCE.

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- New Action: http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE -->
<SQLEXECUTE xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE">
  <SQLSTATEMENT>SELECT tid_seq.nextval id FROM dual</SQLSTATEMENT>
</SQLEXECUTE>
```

The SQLEXECUTE can specify a parameter schema element and a parameter block that contains multiple sets of parameter data. The message shown is for a single invocation of the specified SQL statement so the elements that specify the parameter schema and parameter block are omitted from the message body. For information about the message schema for the SQLEXECUTE operation, see [Message Schemas for the SQLEXECUTE Operation](#).

Specifying the SQLEXECUTE Action

You must specify an action for the message. The following code excerpt shows how to specify the action for the SQLEXECUTE message.

```
Message messageIn = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE", readerIn);
```

Sending the SQLEXECUTE Message

The following code excerpt demonstrates how to invoke a SQLEXECUTE operation on an Oracle database over a channel.

```
// Create Endpoint
EndpointAddress address = new EndpointAddress("oracledb://ADAPTER");

// Create Binding
OracleDBBinding binding = new OracleDBBinding();

// Create Channel Factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, address);
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";
factory.Open();

// Create Request Channel
IRequestChannel channel = factory.CreateChannel();
channel.Open();

// Send Request
System.Xml.XmlReader readerIn = System.Xml.XmlReader.Create("SQLExecute.xml");

Message messageIn = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE", readerIn);
Message messageOut = channel.Request(messageIn);

// Get Response XML
XmlReader readerOut = messageOut.GetReaderAtBodyContents();

// Get tid_seq SEQUENCE
string id = null;
```

```
XmlDocument doc = new XmlDocument();
doc.Load(readerOut);
XmlNodeList list = doc.GetElementsByTagName("ColumnValue");
if (list.Count > 0) id = list[0].InnerXml;
```

Note

The SQLEXECUTE operation always returns a weakly-typed result set.

See Also

Concepts

[Creating a Channel](#)

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Performing an Insert Operation by Using the WCF Channel Model

This section shows how to insert a record into an Oracle database by using a channel. You must specify both a message body and a message action when you send a message.

The Insert Message

The following XML shows a message body for an Insert operation on the HR.EMPLOYEES table. The record set consists of a single employee record. For more information about the schema of an Insert message, see [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#). This is the contents of the Employee_Insert.xml file used in the example.

```
<!-- New namespace: http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEES -->
<Insert xmlns="http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEES">
  <RECORDSET xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <EMPLOYEESRECORDINSERT>
      <EMPLOYEE_ID>0</EMPLOYEE_ID>
      <FIRST_NAME>Anton</FIRST_NAME>
      <LAST_NAME>Kirilov</LAST_NAME>
      <EMAIL></EMAIL>
      <PHONE_NUMBER>555-0198</PHONE_NUMBER>
      <HIRE_DATE>2007-03-01T00:00:00.0000000</HIRE_DATE>
      <JOB_ID>FI_ACCOUNT</JOB_ID>
      <SALARY>5000</SALARY>
      <COMMISSION_PCT>0.15</COMMISSION_PCT>
      <MANAGER_ID>108</MANAGER_ID>
      <DEPARTMENT_ID>100</DEPARTMENT_ID>
    </EMPLOYEESRECORDINSERT>
  </RECORDSET>
</Insert>
```

Specifying the Message Action

You must specify a message action when you send a SOAP message to the Oracle Database adapter. You can specify the message action when you create the message as in the following example.

```
Message messageIn2 = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEES/Insert", readerIn2);
```

The message action in this example, "/HR/Table/EMPLOYEES/Insert", specifies that an Insert operation on the HR.EMPLOYEES table is to be performed

Sending the Insert Message

This example shows how to perform an Insert operation on an Oracle table over a channel. The code uses the SQLEXECUTE operation exposed by the Oracle Database adapter to return the next value of an Oracle SEQUENCE. This value is then written to the EMPLOYEE_ID field in the Insert record. This pattern enables you to insert rows into databases that have an auto-generated primary key value. For more information about invoking the SQLEXECUTE operation over a channel, see [Performing a SQLEXECUTE Operation by Using the WCF Channel Model](#).

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.IO;

using System.ServiceModel;
using System.ServiceModel.Channels;

using Microsoft.ServiceModel.Adapters;
using Microsoft.Adapters.OracleDB;
```

```

namespace OracleDMLChannel
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Endpoint
            EndpointAddress address = new EndpointAddress("oracledb://ADAPTER");

            // Create Binding
            OracleDBBinding binding = new OracleDBBinding();

            // Create Channel Factory
            ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, address);
            factory.Credentials.UserName.UserName = "HR";
            factory.Credentials.UserName.Password = "TIGER";
            factory.Open();

            // Create Request Channel
            IRequestChannel channel = factory.CreateChannel();
            channel.Open();

            // Send Request
            System.Xml.XmlReader readerIn = System.Xml.XmlReader.Create("SQLExecute.xml");

            Message messageIn = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE", readerIn);
            Message messageOut = channel.Request(messageIn);

            // Get Response XML
            XmlReader readerOut = messageOut.GetReaderAtBodyContents();

            // Get Employee ID
            string id = null;
            XmlDocument doc = new XmlDocument();
            doc.Load(readerOut);
            XmlNodeList list = doc.GetElementsByTagName("ColumnValue");
            if (list.Count > 0) id = list[0].InnerText;

            // Compose Insert XML
            XmlDocument insertDoc = new XmlDocument();
            insertDoc.Load("Employee_Insert.xml");

            // Change Employee ID
            XmlNodeList empidList = insertDoc.GetElementsByTagName("EMPLOYEE_ID");
            XmlNode empidNode = empidList[0];
            empidNode.InnerText = id;

            // Change email
            XmlNodeList emailList = insertDoc.GetElementsByTagName("EMAIL");
            XmlNode emailNode = emailList[0];
            emailNode.InnerText = "scotty" + id + "@microsoft.com";

            // Change date
            XmlNodeList dateList = insertDoc.GetElementsByTagName("HIRE_DATE");
            XmlNode dateNode = dateList[0];
            dateNode.InnerText = "2007-03-01T00:00:00.0000000";

            StringReader strReader = new StringReader(insertDoc.InnerText);
            XmlReader readerIn2 = XmlReader.Create(strReader);

            // Send XML
            Message messageIn2 = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/HR/Table/EMPLOYEES/Insert ", readerIn2);
            Message messageOut2 = channel.Request(messageIn2);
        }
    }
}

```

```
        // Close the messages
        messageOut.Close();
        messageOut2.Close();

        channel.Close();
    }
}
```

See Also

Concepts

[Creating a Channel](#)

[Performing a SQLEXECUTE Operation by Using the WCF Channel Model](#)

[Invoking a Function by Using the WCF Channel Model](#)

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Invoking a Function by Using the WCF Channel Model

This section demonstrates how to execute a function in an Oracle database using the channel created in [Creating a Channel](#).

Executing a Function Using the Channel

You can execute a function on an Oracle database by passing an XML message to Microsoft BizTalk Adapter for Oracle Database. The input XML resembles the following:

```
<CREATE_ACCOUNT xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG" xmlns:ns0="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT">
  <REC xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG">
    <ns0:ID>1</ns0:ID>
    <ns0:NAME>Scott</ns0:NAME>
    <ns0:BANKNAME>CitiBank</ns0:BANKNAME>
    <ns0:BRANCH>NY</ns0:BRANCH>
    <ns0:ENABLED>Y</ns0:ENABLED>
  </REC>
</CREATE_ACCOUNT>
```

The following code excerpt demonstrates how to execute a function in an Oracle database using a channel.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;

using System.ServiceModel;
using System.ServiceModel.Channels;

using Microsoft.ServiceModel.Adapters;
using Microsoft.Adapters.OracleDB;

namespace OraclePackageChannel
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Endpoint
            EndpointAddress address = new EndpointAddress("oracledb:// ADAPTER");

            // Create Binding
            OracleDBBinding binding = new OracleDBBinding();

            // Create Channel Factory
            ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, address);
            factory.Credentials.UserName.UserName = "SCOTT";
            factory.Credentials.UserName.Password = "TIGER";
            factory.Open();

            // Create Request Channel
            IRequestChannel channel = factory.CreateChannel();
            channel.Open();

            // Send Request
            System.Xml.XmlReader readerIn = System.Xml.XmlReader.Create("Create_Account.xml");

            Message messageIn = Message.CreateMessage(MessageVersion.Default, "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT", readerIn);
            Message messageOut = channel.Request(messageIn);
        }
    }
}
```

```
// Get Response XML
XmlReader readerOut = messageOut.GetReaderAtBodyContents();

// Get Employee ID
XmlDocument doc = new XmlDocument();
doc.Load(readerOut);
doc.Save("d:\\out.xml");

messageOut.Close();
channel.Close();
}
}
}
```

See Also

Concepts

[Performing an Insert Operation by Using the WCF Channel Model](#)

[Performing a SQLEXECUTE Operation by Using the WCF Channel Model](#)

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Receiving Polling-based Data-changed Messages by Using the WCF Channel Model

You can configure the Microsoft BizTalk Adapter for Oracle Database to poll an Oracle database table or view for any data changes. To perform such a polling operation, the adapter periodically executes a SQL query against an Oracle table or view followed by an optional PL/SQL code block. The results of the SQL query are then returned by the Oracle Database adapter to your code as a strongly-typed result set in an inbound POLLINGSTMT operation. For more information about the mechanism used to configure and perform polling on an Oracle database using the Oracle Database adapter, see [Receiving Polling-based Data-changed Messages](#). It is strongly recommended that you read this topic before proceeding.

You configure the Oracle Database adapter to poll an Oracle database table or view by setting binding properties on an instance of **OracleDBBinding**. In the WCF channel model, you then use this binding to build a channel listener from which you can get an **InputChannel** channel to receive the POLLINGSTMT operation from the adapter.

For an overview of how to receive operations using an **InputChannel** in WCF, see "Service Channel-Level Programming" at <http://go.microsoft.com/fwlink/?LinkId=106053>.

The sections in this topic provide information to help you perform polling on Oracle database tables and views using the WCF channel model.

Consuming the POLLINGSTMT request message

The adapter invokes the POLLINGSTMT operation on your code to poll the Oracle database. That is, the adapter sends a POLLINGSTMT request message that you receive over an **InputChannel** channel shape. The POLLINGSTMT request message contains the result set of the query specified by the **PollingStatement** binding property. You can consume the POLLINGSTMT message in one of two ways:

- To consume the message using node-value streaming you must call the **WriteBodyContents** method on the response message and pass it an **XmlDictionaryWriter** that implements node-value streaming.
- To consume the message using node streaming you can call **GetReaderAtBodyContents** on the response message to get an **XmlReader**.

You typically use node-value streaming to consume result sets that contain Oracle LOB data columns.

For more information about the message structure of the POLLINGSTMT operation, see [Message Schemas for the Polling Operations](#).

For more information about how the Oracle Database adapter supports streaming on LOB data, see [Streaming and the Oracle Database Adapter](#).

For more information about implementing node-value streaming in your code to support end-to-end streaming of LOB data, see [Streaming Oracle LOB Data Types by Using the WCF Channel Model](#).

About the Examples Used in this Topic

The example in this topic uses the SCOTT.ACCOUNTACTIVITY table and the SCOTT.ACCOUNT_PKG.PROCESS_ACTIVITY function. A script to generate these artifacts is supplied with the samples. The example performs the following operations:

- As part of the polling statement, selects all the records from the ACCOUNTACTIVITY table and displays on the console.
- As part of the post poll statement, the example invokes the PROCESS_ACTIVITY function that moves all the records from ACCOUNTACTIVITY table to ACTIVITYHISTORY table.
- Subsequent polls on the ACCOUNTACTIVITY table do not return any records. However, if you want the example to return more records as part of the polling operation, you must insert some records in the ACCOUNTACTIVITY table. You can do so by running the more_activity_data.sql script provided with the samples.

For more information about the samples, see [Samples](#).

How Do I Poll an Oracle Database Using an InputChannel?

To poll an Oracle database table or view to receive data-change messages using the WCF channel model, perform the following steps.

To receive data-changed messages using an `IInputChannel`

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleDB`, `Microsoft.ServiceModel.Channels`, `System.ServiceModel`, and `System.Runtime.Serialization`.
3. Open the Program.cs file and add the following namespaces:
 - `Microsoft.Adapters.OracleDB`
 - `Microsoft.ServiceModel.Channels`
 - `System.ServiceModel`
 - `System.ServiceModel.Description`
 - `System.ServiceModel.Channels`
 - `System.Xml`
 - `System.Runtime.Serialization`
 - `System.IO`
 - `Microsoft.ServiceModel.Channels.Common`
4. Create an instance of **OracleDBBinding** and set the binding properties required to configure polling. At a minimum you must set the **InboundOperationType**, **PollingStatement**, and **PollingInterval** binding properties. For this example, you also set the **PostPollStatement** binding property. For more information about binding properties used to configure polling, see [Receiving Polling-based Data-changed Messages](#).

```
OracleDBBinding binding = new OracleDBBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PollingInterval = 30;
binding.PollingStatement = "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE";
binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;"
```

5. Create a binding parameter collection and set the credentials.

```
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.UserName = "SCOTT";
credentials.UserName.Password = "TIGER";

BindingParameterCollection bindingParams = new BindingParameterCollection();
bindingParams.Add(credentials);
```

6. Create a channel listener and open it. You create the listener by invoking **BuildChannelListener<IInputChannel>** method on the **OracleDBBinding**. You can modify the target namespace for the POLLINGSTMT operation by setting the `PollingId` property in the connection URI. For more information about the adapter connection URI, see [The Oracle Database Connection URI](#).

```
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>
(connectionUri, bindingParams);
```

```
listener.Open();
```

7. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on the listener and open it.

```
IInputChannel channel = listener.AcceptChannel();  
channel.Open();
```

8. Invoke **Receive** on the channel to get the next POLLINGSTMT message from the adapter.

```
Message message = channel.Receive();
```

9. Consume the result set returned by the POLLINGSTMT operation. You can consume the message using either an **XmlReader** or an **XmlDictionaryWriter**.

```
XmlReader reader = message.GetReaderAtBodyContents();
```

10. Close the channel when you have completed processing the request.

```
channel.Close()
```

◆ Important

You must close the channel after you have finished processing the POLLINGSTMT operation. Failure to close the channel may affect the behavior of your code.

11. Close the listener when you are finished receiving data-changed messages.

```
listener.Close()
```

◆ Important

Closing the listener does not close channels created using the listener. You must explicitly close each channel created using the listener.

Example

The following example shows how to configure the Oracle Database adapter to poll Oracle database tables and views and receive the POLLINGSTMT operation using the WCF channel model. The result set returned in the POLLINGSTMT operation is written to the console by using an **XmlReader**.

```
using System;  
using System.Collections.Generic;  
using System.Text;  
  
// Add WCF, WCF LOB Adapter SDK, and Oracle Database adapter namespaces  
using System.ServiceModel;  
using System.ServiceModel.Description;  
using Microsoft.ServiceModel.Channels;  
using Microsoft.Adapters.OracleDB;  
  
// Add this namespace for channel model  
using System.ServiceModel.Channels;  
  
using System.Xml;  
using System.Runtime.Serialization;  
using System.IO;  
  
// Include this namespace for the WCF LOB Adapter SDK and Oracle exceptions
```

```

using Microsoft.ServiceModel.Channels.Common;

namespace OraclePollingCM
{
    class Program
    {
        static void Main(string[] args)
        {
            Uri connectionUri = new Uri("oracleDB://ADAPTER/");

            IChannelListener<IInputChannel> listener = null;
            IInputChannel channel = null;

            // set timeout to receive POLLINGSTMT message
            TimeSpan messageTimeout = new TimeSpan(0, 0, 30);

            Console.WriteLine("Sample Started");

            try
            {
                // Create a binding: specify the InboundOperationType, PollingInterval (in
seconds), the
                // PollingStatement, and the PostPollStatement.
                OracleDBBinding binding = new OracleDBBinding();
                binding.InboundOperationType = InboundOperation.Polling;
                binding.PollingInterval = 30;
                binding.PollingStatement = "SELECT * FROM ACCOUNTACTIVITY FOR UPDATE";
                binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";

                // Create a binding parameter collection and set the credentials
                ClientCredentials credentials = new ClientCredentials();
                credentials.UserName.UserName = "SCOTT";
                credentials.UserName.Password = "TIGER";

                BindingParameterCollection bindingParams = new BindingParameterCollection()
;
                bindingParams.Add(credentials);

                Console.WriteLine("Opening listener");
                // get a listener from the binding
                listener = binding.BuildChannelListener<IInputChannel>(connectionUri, bindi
ngParams);
                listener.Open();

                Console.WriteLine("Opening channel");
                // get a channel from the listener
                channel = listener.AcceptChannel();
                channel.Open();

                Console.WriteLine("Channel opened -- waiting for polled data");
                Console.WriteLine("Receive request timeout is {0}", messageTimeout);

                // Poll five times with the specified message timeout
                // If a timeout occurs polling will be aborted
                for (int i = 0; i < 5; i++)
                {
                    Console.WriteLine("Polling: " + i);
                    Message message = null;
                    XmlReader reader = null;
                    try
                    {
                        //Message is received so process the results
                        message = channel.Receive(messageTimeout);
                    }
                    catch (System.TimeoutException toEx)
                    {
                        Console.WriteLine("\nNo data for request number {0}: {1}", i + 1, t

```

```
oEx.Message);
```

```
        continue;
    }

    // Get the query results using an XML reader
    try
    {
        reader = message.GetReaderAtBodyContents();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception : " + ex);
        throw;
    }
}
```

```
Console
```

```
);
```

```
    // Write the TID, ACCOUNT, AMOUNT, and TRANSDATE for each record to the
    Console.WriteLine("\nPolling data received for request number {0}", i+1
    Console.WriteLine("Tx ID\tACCOUNT\tAMOUNT\tTx DATE");
```

```
while (reader.Read())
{
    if (reader.IsStartElement())
    {
        switch (reader.Name)
        {
            case "POLLINGSTMTRECORD":
                Console.WriteLine("\n");
                break;

            case "TID":
                reader.Read();
                Console.WriteLine(reader.ReadString() + "\t");
                break;

            case "ACCOUNT":
                reader.Read();
                Console.WriteLine(reader.ReadString() + "\t");
                break;

            case "AMOUNT":
                reader.Read();
                Console.WriteLine(reader.ReadString() + "\t");
                break;

            case "TRANSDATE":
                reader.Read();
                Console.WriteLine(reader.ReadString() + "\t");
                break;

            default:
                break;
        }
    }
}
```

```
    // return the cursor
    Console.WriteLine();
```

```
    // close the reader
    reader.Close();
```

```
de above with the following
```

```
    //          To save the polling data to a file you can REPLACE the co
    //
    //          XmlDocument doc = new XmlDocument();
    //          doc.Load(reader);
    //          using (XmlWriter writer = XmlWriter.Create("PollingOutput
```


Streaming Oracle LOB Data Types by Using the WCF Channel Model

The Microsoft BizTalk Adapter for Oracle Database supports end-to-end streaming of LOB data for certain operations. The sections in this topic describe how to implement streaming for LOB data when you use the WCF channel model.

For background information about how the adapter supports streaming of LOB data types, see [Streaming and the Oracle Database Adapter](#). You should read this topic before proceeding.

A sample that demonstrates LOB data streaming is available in the SDK samples included with the Oracle Database adapter. For more information, see [Samples](#).

Streaming Outbound Messages to the Adapter

The adapter supports end-to-end LOB data streaming for the request message for the UpdateLOB operation.

To support end-to-end streaming on UpdateLOB operations in the WCF channel model, you must:

1. Set the **UseAmbientTransaction** binding property to true.
2. Implement a **System.ServiceModel.Channels.BodyWriter** that is capable of streaming the LOB data (performing node-value streaming on the LOB data).
3. Perform the UpdateLOB operation within a transaction scope.
4. Create the **System.ServiceModel.Message** used to invoke the operation by supplying the message body with this **BodyWriter** using an appropriate overload of the **Message.Create** method.

Setting the UseAmbientTransaction Binding Property

The following example shows how to create a binding for the Oracle Database adapter and set the **UseAmbientTransaction** binding property.

```
// Create binding
OracleDBBinding odbBinding = new OracleDBBinding();

//set the binding property
binding.UseAmbientTransaction = true;
```

Implementing a BodyWriter

The following example shows an implementation of a **BodyWriter** that performs node-value streaming.

```
/// <summary>
/// This class overrides the OnWriteBodyContents function to do node-value streaming
/// </summary>
class StreamingBodyWriter : BodyWriter, IDisposable
{
    XmlReader m_reader = null;

    int m_chunkSize;
    /// <summary>
    /// Initializes the body writer
    /// </summary>
    /// <param name="reader">Reader for input</param>
    /// <param name="chunkSize">The chunksize in which the data is passed to adapter</param>
    >
    public StreamingBodyWriter(XmlReader reader, int chunkSize)
        : base(false)
    {
        m_reader = reader;
```

```

        if (chunkSize <= 0)
            throw new ApplicationException("ChunkSize should be a positive value");
        m_chunkSize = chunkSize;
    }

protected override void OnWriteBodyContents(XmlDictionaryWriter writer)
{
    if (m_reader == null)
        throw new ApplicationException("Reader cannot be null");

    while (m_reader.Read())
    {
        switch (m_reader.NodeType)
        {
            case XmlNodeType.Element:
                writer.WriteStartElement(m_reader.LocalName, m_reader.NamespaceURI);
                break;
            case XmlNodeType.Text:
                #region Streaming Code
                char[] tempBuffer = new char[m_chunkSize];
                int length = 0;
                while ((length = m_reader.ReadValueChunk(tempBuffer, 0, m_chunkSize)) >
0)
                {
                    writer.WriteString(new String(tempBuffer, 0, length));
                }
                #endregion
                break;
            case XmlNodeType.EndElement:
                writer.WriteEndElement();
                break;
        }
    }
}

#region IDisposable Members

public void Dispose()
{
    if (m_reader != null)
    {
        m_reader.Close();
        m_reader = null;
    }
}

#endregion
}

```

Perform the Operations Within a Transaction Scope

The following example shows how to perform operations within a transaction scope.

```

// Create a transaction scope
using(TransactionScope tx = new TransactionScope())
{
    // perform operations within the transaction
    // ...
    // ...

    //Complete the transaction
    tx.Complete()
}

```

Creating a Message by using a BodyWriter

The following example shows how to create an UpdateLOB request message using the **BodyWriter** in the preceding example. The message data is read from a file.

```
// Create a transaction scope
using(TransactionScope tx = new TransactionScope())
{
    XmlReader readerIn = XmlReader.Create ("updatelob.xml");
    // StreamingBodyWriter class is responsible for streaming
    StreamingBodyWriter stBW = new StreamingBodyWriter(readerIn, chunkSize);

    Message InputMsg = Message.CreateMessage(MessageVersion.Default,
        "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/UpdateLOB",
        stBW);

    //Send the request message and get the output message
    OutputMsg = channel.Request(InputMsg);

    tx.Complete();
}
```

Streaming Inbound Messages from the Adapter

The adapter supports end-to-end LOB data streaming for the following inbound messages:

- Response message for functions with OUT or IN OUT parameters that contain LOB data. Note that RECORD TYPE parameters can contain LOB data columns.
- Response message for functions with OUT REF CURSOR parameters (or return values) that contain LOB data. This includes the output side of IN OUT REF CURSOR parameters.
- Response message for procedures with IN or IN OUT parameters that contain LOB data. Note that RECORD TYPE parameters can contain LOB data columns.
- Response message for procedures with OUT REF CURSOR parameters that contain LOB data. This includes the output side of IN OUT REF CURSOR parameters
- Response message for SQLEXECUTE operations that return result sets that contain LOB data.
- Response message for Table or view Select operations that return LOB data in the result set.
- Request message for the (inbound) POLLINGSTMT operation

To support end-to-end streaming on an inbound message in the WCF channel model, you must:

1. Implement a **System.Xml.XmlDictionaryWriter** that is capable of streaming the LOB data (performing node-value streaming on the LOB data).
2. Consume the **Message** by invoking **WriteBodyContents** method with this **XmlDictionaryWriter**.

Implementing an XmlDictionaryWriter

The following example shows an implementation of an **XmlDictionaryWriter** that performs node-value streaming.

```
using System;
using System.Xml;
using System.Text;

class FileXmlWriter : XmlDictionaryWriter
{
    XmlTextWriter xts;
```

```
public FileXmlWriter(string file)
{
    xts = new XmlTextWriter(file, Encoding.UTF8);
}

public override void WriteBase64(byte[] buffer, int index, int count)
{
    xts.WriteBase64(buffer, index, count);
}

public override void WriteCDATA(string text)
{
    xts.WriteCDATA(text);
}

public override void WriteCharEntity(char ch)
{
    xts.WriteCharEntity(ch);
}

public override void WriteChars(char[] buffer, int index, int count)
{
    xts.WriteChars(buffer, index, count);
}

public override void WriteComment(string text)
{
    xts.WriteComment(text);
}

public override void WriteDocType(string name, string pubid, string sysid, string subset)
{
    xts.WriteDocType(name, pubid, sysid, subset);
}

public override void WriteEndAttribute()
{
    xts.WriteEndAttribute();
}

public override void WriteEndDocument()
{
    xts.WriteEndDocument();
}

public override void WriteEndElement()
{
    xts.WriteEndElement();
}

public override void WriteEntityRef(string name)
{
    xts.WriteEntityRef(name);
}

public override void WriteFullEndElement()
{
    xts.WriteFullEndElement();
}

public override void WriteProcessingInstruction(string name, string text)
{
    xts.WriteProcessingInstruction(name, text);
}

public override void WriteRaw(string data)
```

```

{
    xts.WriteRaw(data);
}

public override void WriteRaw(char[] buffer, int index, int count)
{
    xts.WriteRaw(buffer, index, count);
}

public override void WriteStartAttribute(string prefix, string localName, string ns)
{
    xts.WriteStartAttribute(prefix, localName, ns);
}

public override void WriteStartDocument(bool standalone)
{
    xts.WriteStartDocument(standalone);
}

public override void WriteStartDocument()
{
    xts.WriteStartDocument();
}

public override void WriteStartElement(string prefix, string localName, string ns)
{
    xts.WriteStartElement(localName);
}

public override void WriteString(string text)
{
    xts.WriteString(text);
}

public override void WriteSurrogateCharEntity(char lowChar, char highChar)
{
    xts.WriteSurrogateCharEntity(lowChar, highChar);
}

public override void WriteWhitespace(string ws)
{
    xts.WriteWhitespace(ws);
}

public override void Close()
{
    xts.Close();
}

public override void Flush()
{
    xts.Flush();
}

public override string LookupPrefix(string ns)
{
    return xts.LookupPrefix(ns);
}

public override WriteState WriteState
{
    get { return xts.WriteState; }
}
}

```

The following example shows how to consume a table Select response message using the **FileXmlWriter** implemented in the preceding example. (The **FileWriter** class was created by sub-classing **XmlDictionaryWriter**.) The example uses an **IRequestChannel** channel to invoke the Select operation. The details of the channel creation have been omitted. The Select request message is read from a file and the Select response message is written to a file.

```
// Read Select message body from a file
XmlReader readerIn = XmlReader.Create("select.xml");
Message InputMsg = Message.CreateMessage(MessageVersion.Default,
    "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/Select", readerIn);

Message OutputMsg = channel.Request(InputMsg);

// Streaming response message to select_output.xml using the custom XmlDictionaryWriter;
FileXmlWriter fileXmlWriter = new FileXmlWriter("select_output.xml");
OutputMsg.WriteBodyContents(fileXmlWriter);
fileXmlWriter.Flush();
fileXmlWriter.Close();

// Streaming complete close output message;
OutputMsg.Close();
```

The following XML shows the request message (contents of the select.xml file) for the Select operation. The CUSTOMER table contains a BLOB column named PHOTO.

```
<Select xmlns="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER">
  <COLUMN_NAMES>*</COLUMN_NAMES>
  <FILTER>NAME='Kim Ralls'</FILTER>
</Select>
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Retrieving Metadata Programmatically from the Oracle Database

The Microsoft BizTalk Adapter for Oracle Database is a custom WCF binding that exposes an Oracle database as a WCF service. The adapter exposes the Oracle database as a self-describing service; that is, a service that is capable of publishing metadata about the operations that it supports. Metadata describes the logical interface to a WCF service; that is, the service contract, messages, and message schemas that must be used to interact with the service.

This metadata is used by tools such as:

- The Add Adapter Service Reference Visual Studio Plug-in to generate managed code representations of the service contract, and
- The Consume Adapter Service BizTalk Project Add-in to generate message schemas.

However, you can also retrieve metadata programmatically from the adapter. For example, you might want to do this to create a custom metadata retrieval tool to use in an existing application.

The adapter publishes metadata through two endpoints:

- A WS-Metadata Exchange (MEX) endpoint. WCF automatically provides a MEX endpoint for all WCF bindings. You can use metadata exchange to retrieve metadata for operations supported by the adapter on the underlying Oracle database.
- An **IMetadataRetrievalContract** endpoint. The **IMetadataRetrievalContract** interface is implemented by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK. It categorizes Oracle database artifacts at multiple logical levels and presents them as a tree of metadata nodes. You can use methods exposed by the **IMetadataRetrievalContract** interface to browse and search the nodes of this tree and to return metadata for operations in which you are interested.

The topics in this section describe how to use MEX and **IMetadataRetrievalContract** endpoints to retrieve metadata programmatically from the adapter.

In This Section

- [Retrieving Metadata Using WS-Metadata Exchange](#)
- [Retrieving Metadata Using IMetadataRetrievalContract](#)

See Also

Concepts

[Development](#)

Retrieving Metadata Using WS-Metadata Exchange

As a Windows Communication Foundation (WCF) custom binding, the Microsoft BizTalk Adapter for Oracle Database exposes a WS-Metadata Exchange (MEX) endpoint that you can use to retrieve metadata for specific operations from the Microsoft BizTalk Adapter for Oracle Database.

WCF provides a rich infrastructure for exporting, publishing, retrieving and importing metadata about a service. WCF services, like the adapter, use metadata to describe how to interact with the service endpoints so that tools, like svcutil.exe, can automatically generate client code for consuming the service. WCF represents the metadata for a service as an instance of the **MetadataSet** type, which is strongly tied to the metadata serialization format defined in WS-Metadata Exchange (MEX). You can create a **MetadataSet** for targeted operations on the adapter by using a **MetadataExchangeClient**.

WCF support for metadata exchange is an expansive topic and beyond the scope of this documentation. For more information about support for metadata in WCF, see "Metadata" in the WCF documentation at <http://go.microsoft.com/fwlink/?LinkId=105634>. For a particularly good description of the architecture, classes, and namespaces that WCF exposes for metadata, see "Metadata Architecture Overview" at <http://go.microsoft.com/fwlink/?LinkId=105635>. You should familiarize yourself with the content related to retrieving metadata from a WCF service in these WCF topics before proceeding.

The following topics contain information about how to use a **MetadataExchangeClient** to retrieve metadata from the Oracle Database adapter.

Using a MetadataExchangeClient to Retrieve Metadata

To use a **MetadataExchangeClient** you must specify a connection URI and a binding (**OracleDBBinding**). The connection URI identifies the operations for which you want to retrieve metadata.

The following sections contain information about how to specify the connection URI, important binding properties, and how to use a **MetadataExchangeClient** to retrieve metadata from the adapter.

The Connection URI

To use the **MetadataExchangeClient** you must supply an Oracle connection URI that specifies a MEX endpoint and the operation or operations for which you want to retrieve metadata. You specify a MEX endpoint and target operations in the connection URI in the following manner:

- You must include the "wsdl" parameter in the query string. If it is the first parameter in the query string, it is specified just after the question mark (?). If it is not the first parameter, it should be preceded with an ampersand (&).
- You must follow the "wsdl" parameter by one or more "op" parameters. Each "op" parameter is preceded by an ampersand (&) and specifies the message action (node ID) of a target operation.

For example, the following connection URI targets the Insert and Delete operations for the SCOTT.EMP table. The "wsdl" and "op" parameters are highlighted.

```
"oracledb://ADAPTER?wsdl&op=http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert&op=http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete"
```

Note

If you want to modify the namespace generated for the POLLINGSTMT operation you should specify a PollingId parameter in the query string.

How you pass this connection URI to the **MetadataExchangeClient** depends on which of the overloaded methods you use to create the client and retrieve metadata from the adapter.

For more information about the Oracle connection URI, see [The Oracle Database Connection URI](#).

Binding Properties

When you create the **MetadataExchangeClient**, you must specify an **OracleDBBinding**.

There are several binding properties that affect how the adapter generates metadata. These properties are:

- **EnableSafeTyping**
- **UseSchemaInNamespace**
- **PollingStatement**

◆ Important

If you want to retrieve metadata for the POLLINGSTMT operation you must set the **PollingStatement** binding property.

You should ensure that these binding properties are set to the values required for your application before you invoke the **GetMetadata** method on the **MetadataExchangeClient**. For more information about the Oracle Database adapter binding properties, see Working with [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Example

The following example uses a **MetadataExchangeClient** to create a service description (WSDL document) for the Insert, Update, Delete, and Select operations on the SCOTT.EMP table. The WSDL is saved to a file, EmpOperations.wsdl.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Collections.ObjectModel;

// Needed for WCF and Oracle Adapter
using System.ServiceModel;
using Microsoft.ServiceModel.Channels;
using Microsoft.Adapters.OracleDB;

// Needed for MetadataExchangeClient class
using System.ServiceModel.Description;
// Needed for ServiceDescription class
using System.Web.Services;

namespace OracleMetadataExchange
{
    class Program
    {
        static void Main(string[] args)
        {
            //create a binding
            OracleDBBinding binding = new OracleDBBinding();

            //create a metadata exchange client that will retrieve metadata according to th
            e WS-MEX standard
            MetadataExchangeClient client = new MetadataExchangeClient(binding);
            client.SoapCredentials.UserName.UserName = "SCOTT";
            client.SoapCredentials.UserName.Password = "TIGER";

            //set up an endpoint address and specifies the operations for which we want met
            adata
            string connectionUri = "oracledb://ADAPTER?wsdl"
                + "&op="
                + "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert"
                + "&op="
                + "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Update"
                + "&op="
                + "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete"
                + "&op="
                + "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select";

            EndpointAddress address = new EndpointAddress(connectionUri);

            //get the metadata
            MetadataSet ms = client.GetMetadata(address);
        }
    }
}
```

```
// Check for the metadata set size
Collection<MetadataSection> documentCollection = ms.MetadataSections;
if (documentCollection != null && documentCollection.Count > 0)
{
    //get the wsdl from the metadata set
    System.Web.Services.Description.ServiceDescription wsdl = (System.Web.Services.Description.ServiceDescription)documentCollection[0].Metadata;

    //save the wsdl to a file
    wsdl.Write("EmpOperations.wsdl");
}
}
}
```

See Also

Other Resources

[Retrieving Metadata Programmatically from the Oracle Database](#)

Retrieving Metadata Using IMetadataRetrievalContract

The Microsoft BizTalk Adapter for Oracle Database exposes an **IMetadataRetrievalContract** endpoint that you can use to browse and search for Oracle database artifacts and to retrieve metadata for operations in the form of a Web Services Description Language (WSDL) document.

The **IMetadataRetrievalContract** interface is implemented by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and provides metadata browse, search, and retrieval capabilities. In addition to the **IMetadataRetrievalContract** interface, the WCF LOB Adapter SDK exposes the **MetadataRetrievalClient** class, which implements the interface. You can use either an **IMetadataRetrievalContract** channel or a **MetadataRetrievalClient** to work with metadata; the methods exposed to browse, search, and retrieve metadata are the same in each case.

The following sections provide information about how to use the **IMetadataRetrievalContract** interface.

The IMetadataRetrievalContract Interface

The following table provides information about important classes that are used when you work with the **IMetadataRetrievalContract** interface.

Class or Interface	Description
IMetadataRetrievalContract interface (Microsoft.ServiceModel.Channels)	Defines the Browse , Search , and GetMetadata methods. You invoke these methods either by using an IMetadataRetrievalContract channel or a MetadataRetrievalClient to work with adapter metadata.
MetadataRetrievalClient class (Microsoft.ServiceModel.Channels)	Implements the IMetadataRetrievalContract interface. You can create an instance of this class and configure it for your Oracle database by providing an OracleDBBinding and an EndpointAddress . Then you can invoke its methods to work with metadata.
MetadataRetrievalNode class (Microsoft.ServiceModel.Channels)	Represents a metadata node on the adapter. The Browse and Search methods return nodes of this type, and the GetMetadata method takes nodes of this type as a parameter.
ServiceDescription class (System.Web.Services.Description)	Provides a means of creating and formatting a valid WSDL document file. The GetMetadata method returns a ServiceDescription object.

For more information about the **IMetadataRetrievalContract** interface, the **MetadataRetrievalClient** class, and the **MetadataRetrievalNode** class; see the **Microsoft.ServiceModel.Channels** managed reference at <http://go.microsoft.com/fwlink/?LinkId=105566>.

Metadata Node IDs

The adapter organizes its metadata as a hierarchical tree of nodes. Within this tree structure there are two types of metadata nodes:

- **Operation nodes** represent operations that the adapter surfaces on Oracle database artifacts. Operation nodes are the leaves of the tree.
- **Category nodes** represent Oracle database artifacts and groupings of Oracle database artifacts that do not directly

correspond to an operation on the adapter. Category nodes are the branches of the tree; they contain other category nodes and/or operation nodes. For example, Oracle tables and packages are represented as category nodes.

Each metadata node surfaced by the adapter is identified by a unique node ID. For more information about the metadata node IDs surfaced by the adapter, see [Metadata Node IDs](#). You use these node IDs to specify target Oracle database artifacts when you use the **IMetadataRetrievalContract** interface to browse, search, and retrieve metadata.

Binding Properties

Whether you use an **IMetadataRetrievalContract** channel or an **IMetadataRetrievalClient** to work with metadata, you must specify an **OracleDBBinding** when you create the instance.

There are several binding properties that affect how the adapter generates metadata. These properties are:

- **EnableSafeTyping**
- **UseSchemaInNamespace**
- **PollingStatement**

◆ Important

If you want to retrieve metadata for the POLLINGSTMT operation you must set the **PollingStatement** binding property.

You should ensure that these binding properties are set to the values required for your application before you open the metadata retrieval object. For more information about the Oracle Database adapter binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Browsing Metadata Nodes

You use the **Browse** method to return all the metadata nodes that are contained in a parent node. The following example browses for the first three schemas on the Oracle database. In this example, **client** is an instance of **MetadataRetrievalClient**.

```
// The first parameter is the node ID.  
// The second parameter is the start index.  
// The third parameter is the maximum number of nodes to return.  
MetadataRetrievalNode[] nodes = client.Browse(MetadataRetrievalNode.Root.No  
deId, 0, 3);
```

◆ Important

You can only browse category nodes; you cannot browse operation nodes.

Searching for Metadata Nodes

You use the **Search** method to perform a search for nodes contained by a parent node. The adapter supports wildcard characters in search expressions; for example you can specify the percent (%) wildcard character to match zero or more characters. The following example shows a search for all the tables in the SCOTT schema that contain the string "EMP". In this example, **client** is an instance of **MetadataRetrievalClient**.

```
// Search for all nodes that contain "EMP" under the SCOTT.Table node.  
// The parameters are the parent node ID, the search expression, and  
// the maximum number of nodes to return.  
IMetadataRetrievalNode[] nodes = client.Search("http://Microsoft.LobServices.OracleDB/2007/  
03/SCOTT/Table", "%EMP%", 3);
```

◆ Important

Searching is only supported on a limited set of nodes. For more information about the nodes on which search is supported and about the wildcard characters supported in search expressions, see [Metadata Node IDs](#).

Retrieving Metadata (WSDL) for Operations

You use the **GetMetadata** method to retrieve a service description (WSDL document) for a group of operation nodes. The following example retrieves a service description that contains all of the operations that the adapter surfaces for the SCOTT.EMP table by specifying its node ID. In this example, **client** is an instance of **MetadataRetrievalClient**.

```
// Get a service description that contains all of the operations
// surfaced for the SCOTT.EMP table. The IsOperation
// property is set false because this is a category node.
nodes = new MetadataRetrievalNode[1];
nodes[0] = new MetadataRetrievalNode();
nodes[0].NodeId = "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP";
nodes[0].IsOperation = false;
System.Web.Services.Description.ServiceDescription description = client.GetMetadata(nodes);
```

◆ Important

The **IsOperation** property should be false for category nodes and true for operation nodes.

Using a MetadataRetrievalClient

Creating and using a **MetadataRetrievalClient** is much the same as any other WCF client. You create the client by specifying an endpoint and an instance of **OracleDBBinding**. You can do this either declaratively in configuration or imperatively in your code. You then invoke the methods of the **MetadataRetrievalClient** to browse, search, and retrieve metadata from the adapter.

The following example shows how to use a **MetadataRetrievalClient** to browse, search, and retrieve metadata from the Oracle Database adapter. The example demonstrates:

- Browsing the root node of the metadata tree for Oracle Database schemas.
- Searching for the tables in the SCOTT schema with names that contain the string "EMP".
- Retrieving metadata for all of the operations supported for the SCOTT.EMP table by passing a category node to the **GetMetadata** method.
- Retrieving metadata for the POLLINGSTMT operation by passing the POLLINGSTMT operation node to the **GetMetadata** method..

```
using System;
using System.Collections.Generic;
using System.Text;

using System.ServiceModel;
using Microsoft.Adapters.OracleDB;
using Microsoft.ServiceModel.Channels;

using System.Web.Services.Description;

namespace OracleMetadataRetrieval
{
    class NodeWriter
    {
        // This method writes the value of a collection of metadata retrieval nodes
        // to the console
        public void Write(string title, MetadataRetrievalNode[] nodes)
        {
            Console.WriteLine(title);
            Console.WriteLine();
        }
    }
}
```

```

//write all the nodes returned to the console.
foreach (MetadataRetrievalNode node in nodes)
{
    Console.WriteLine("NodeId = " + node.NodeId);
    Console.WriteLine("\tDirection    = " + node.Direction.ToString());
    Console.WriteLine("\tIsOperation = " + node.IsOperation.ToString());
    Console.WriteLine("\tDisplayName = " + node.DisplayName);
    Console.WriteLine("\tDescription = " + node.Description);
}
Console.WriteLine();
}
}

class Program
{
    static void Main(string[] args)
    {
        MetadataRetrievalClient client = null;
        NodeWriter nodeWriter = new NodeWriter();

        try
        {
            // create a binding and
            // set the PollingStatement binding property if you want to
            // return metadata for the POLLINGSTMT operation
            OracleDBBinding binding = new OracleDBBinding();
            binding.PollingStatement = "SELECT * FROM EMP";

            // Set PollingId parameter if you want to alter the namespace of the POLLIN
            // GSTMT operation
            EndpointAddress address = new EndpointAddress("oracledb://ADAPTER?PollingId
            =1");

            client = new MetadataRetrievalClient(binding, address);
            client.ClientCredentials.UserName.UserName = "SCOTT";
            client.ClientCredentials.UserName.Password = "TIGER";
            client.Open();

            // Browse for the first 3 (schema) nodes directly under the root
            // The parameters are the parent Node ID, the start index, and the maximum
            // number
            // of nodes to return
            MetadataRetrievalNode[] nodes = client.Browse(MetadataRetrievalNode.Root.No
            deId, 0, 3);
            nodeWriter.Write("Browse results for the root node:", nodes);

            // Search for first 3 tables that contain "EMP" in the SCOTT schema
            // The parameters are the parent node ID, the search expression, and the ma
            // ximum number
            // of nodes to return
            nodes = client.Search("http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/
            Table", "%EMP%", 3);
            nodeWriter.Write(String.Format("Search results for \"%EMP%\" under {0} node
            :", "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table"), nodes);

            // Get a WSDL document that specifies a contract that contains the operatio
            // ns
            // surfaced for the SCOTT.EMP table. The IsOperation property is set false
            // because this is a category node.
            nodes = new MetadataRetrievalNode[1];
            nodes[0] = new MetadataRetrievalNode();
            nodes[0].NodeId = "http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Tabl
            e/EMP";
            nodes[0].IsOperation = false;
            System.Web.Services.Description.ServiceDescription description = client.Get
            Metadata(nodes);
            description.Write("EmpTableContract.wsdl");
        }
    }
}

```



```
Direction    = Outbound
IsOperation  = False
DisplayName   = EMP1
Description  = Table.EMP1
```

Using an IMetadataRetrievalContract Channel

You can also create an **IMetadataRetrievalContract** channel and then use this channel to browse, search, and retrieve metadata from the adapter. (The method signatures are the same as for the **MetadataRetrievalClient** class.) The following example shows how to do this.

```
...
//Create a binding and endpoint address.
OracleDBBinding binding = new OracleDBBinding();
EndpointAddress address = new EndpointAddress("oracledb://ADAPTER/");

//Create and open a channel factory that will return an IMetadataRetrievalContract object,
on which browse, search, and get can be performed.
ChannelFactory<IMetadataRetrievalContract> factory = new ChannelFactory<IMetadataRetrievalC
ontract>(binding, address);
factory.Credentials.UserName.UserName = "SCOTT";
factory.Credentials.UserName.Password = "TIGER";
factory.Open();

//Obtain an IMetadataRetrievalContract channel from the factory.
IMetadataRetrievalContract channel = factory.CreateChannel();

//Perform a search using the channel.
MetadataRetrievalNode[] nodes = channel.Search("http://Microsoft.LobServices.OracleDB/2007/
03/SCOTT/Table", "%EMP%", int.MaxValue);
...
```

See Also

Other Resources

[Retrieving Metadata Programmatically from the Oracle Database](#)

Using the Oracle Database Adapter with Microsoft Office SharePoint Server

The WCF Adapter Service Development Wizard for Visual Studio 2010 enables the Microsoft BizTalk Adapter for Oracle Database and the Microsoft BizTalk Adapter for Oracle E-Business Suite to be directly consumed as an external datasource in Microsoft Office SharePoint Server 2010. The Add Service Development Wizard that supports this feature is launched with the **WCF Adapter Service** template for creating a new Visual C# Web Sites in Visual Studio 2010. The template is included with the BizTalk Adapter Pack 2010 . You must also install the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK 2010.

SharePoint Operation Support

The Adapter Service Development wizard generates a special service contract for the Oracle adapters that is compatible with Microsoft Office SharePoint Server 2010. The wizard will generate a service contract which includes the following operations for integrating the adapter with Microsoft Office SharePoint Server:

- **Create:** Supported by the CreateItem_ operation.
- **Read:** Supported by the ReadItem_ operation.
- **Update:** Supported by the UpdateItem_ operation.
- **Delete:** Supported by the DeleteItem_ operation.
- **Query:** Supported by the ReadList operation.
- **Associate:** Supported by the Associate_ operation.

The following service contract was generated using for the Microsoft BizTalk Adapter for Oracle Database as an example. The adapter is configured to provide access to the EMP table

```
[System.ServiceModel.ServiceContractAttribute()]
public interface ISCOTT_EMP {

    [System.ServiceModel.OperationContractAttribute()]
    SCOTT_EMP_Record[] ReadList(System.Nullable<int> Limit);

    [System.ServiceModel.OperationContractAttribute()]
    void CreateItem(SCOTT_EMP_Record Input);

    [System.ServiceModel.OperationContractAttribute()]
    SCOTT_EMP_Record[] ReadItem_EMPNO(System.Nullable<decimal> EMPNO);

    [System.ServiceModel.OperationContractAttribute()]
    void UpdateItem_EMPNO(SCOTT_EMP_Record Input);

    [System.ServiceModel.OperationContractAttribute()]
    void DeleteItem_EMPNO(System.Nullable<decimal> EMPNO);

    [System.ServiceModel.OperationContractAttribute()]
    SCOTT_EMP_Record[] Associate_DEPTNO(System.Nullable<decimal> DEPTNO);
}
```

Creating a New Web Site to Host the Microsoft BizTalk Adapter for Oracle Database in IIS

These steps provide an example using the WCF Adapter Service Development Wizard, to create a new WCF web service hosting the Microsoft BizTalk Adapter for Oracle Database. The service contract will include operations directly compatible with Microsoft Office Sharepoint Server 2010. So that it can be directly consumed as an external datasource. The adapter is configured to authenticate with the Oracle database using the **SCOTT** account. If the **SCOTT** account is locked, you can unlock

the account by logging into SQL Plus as SYSDBA.

```
<Oracle Installation Bin Directory>\Sqlplus.exe SYS AS SYSDBA
```

Then run the following command.

```
SQL> ALTER USER scott ACCOUNT UNLOCK;
```

Creating the New Web Site Project

1. Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2010**, and then click **Microsoft Visual Studio 2010**.
2. In Visual Studio 2010, on the **File** menu, select **New** and then click **Project**.
3. In the **New Project** dialog box, expand **Other Languages** and click **Visual C#**. Find the **WCF Adapter Service** in the template list and click it to select it.

Note

The **WCF Adapter Service** template is not available if the BizTalk Adapter Pack 2010 is not installed. On x64 systems, install both the x86 and x64 versions of the BizTalk Adapter Pack 2010 .

4. Specify **ScottEMP** for the name, and then click **OK**. The **WCF Adapter Service Development Wizard** starts.
5. On the **Introduction** page, click **Next**.
6. On the **Choose Operations** page, specify the **oracleDBBinding** binding.
7. Click the **Configure** button. The **Configure Adapter** dialog is displayed.
8. On the **Security** tab, select **Username** in the **Client credential type** dropdown list box.
9. Enter **SCOTT** for the User name and enter the correct password for the SCOTT account. The default password for the SCOTT account is **tiger**.
10. Click the **URI Properties** tab, enter the IP address or host name for your Oracle server in the **ServerAddress** box.
11. Enter the correct Oracle database service instance name in the **ServiceName** box. You can copy the instance name information from Oracle Enterprise Manager.
12. Press the **OK** button on the **Configure Adapter** dialog
13. On the **Choose Operations** page of the wizard, click the **Connect** button and wait a few moments for the categories to be built for the Oracle database.
14. Once the categories are added in the **Select a category** list, scroll down to **SCOTT** and expand it. Then expand **Table** and click the **EMP** table entry.
15. In the **Available categories and operations** list, select all the operations in the list and click the **Add** button. All the operations are added to the **Added categories and operations** list.
16. On the **Choose Operations** page, click the **Next** button.
17. On the **Configure Service and Endpoint Behaviors** page, set the **UseServiceCertificate** Service behavior to **false** for this example. Then click the **Next** button.
18. On the **Configure Service Endpoint Binding and Address** page, click the **Apply** button. Then click the **Next** button.
19. On the **Summary** page, click the **Finish** button.
20. Click the **Build** menu option and then click **Build Solution**. Verify the project build was successful with no errors.

Publishing the New Service to IIS

For this example you will publish the adapter host service to the local IIS web server.

1. In Solution Explorer for Visual Studio 2010, right click the **ScottEmp** project and click **Properties**. The Project Designer

tabs are displayed.

2. Click the **Web** tab, then click the **Use Local IIS Web server** option.
3. Click the **Create Virtual Directory** button.
4. Open a web browser to the service address **http://localhost/ScottEmp/ISCOTT_EMP.svc**. You should receive a message stating "You have created a service" indicating the adapter is hosted in IIS.

Adding the External Data Source to a SharePoint Site using SharePoint Designer

This section describes how to add the WCF Service as an external data source to a new Web Site using SharePoint Designer.

Adding the External Data source

1. Open SharePoint Designer and create a new Web Site.
2. In SharePoint Designer, expand **Navigation** and click **External Content types** in the **Site Objects** list.
3. Click the **External Content Type** menu button to create a new external content type.
4. Click the text beside **Name** to edit the name of the new external content type. Enter **OracleEMP** for the name.
5. Click the text link beside **External System** which says **Click here to discover external data sources and operations..** This opens the Operation Designer for the OracleEMP external content type.
6. Click the **Add Connection** button on the discovery screen.
7. In the External Data Source Type Selection dialog, choose **WCF Service** and click the **OK** button.
8. In the WCF Connection dialog, in the **Service Metadata URL** box enter **https://localhost/ScottEmp/ISCOTT_EMP.svc?wsdl**
9. In the **Service Endpoint URL** box enter **https://localhost/ScottEmp/ISCOTT_EMP.svc**
10. Click the **OK** button to close the WCF Connection dialog.
11. Once the Data Source information is populated, expand the **https://localhost/ScottEmp/ISCOTT_EMP.svc** data source and expand **Web Methods**.
12. Right click the **ReadList** Web Method and click **New Read List Operation**. The Read List configuration dialog is launched.
13. In the Read List dialog click **Return Parameters** and click **EMPNO** in the Data Source Elements. Click the **Map to identifier**.
14. Click **Finish** in the Read List dialog.
15. Save the new external data source by typing **Ctrl+s**.

Testing the External Data Source Connection

1. In the new web site, click the **Create Lists and Forms** button. The Create List and Form for OracleEMP dialog appears.
2. Enter **OracleEMP_List** for the List Name and click the **OK** button.
3. Once the list is create, click the **Summary View** button on the menu.
4. Click **OracleEMP_List** under External Lists.
5. Click the **Preview in Browser** button on the menu to test the ReadList operation of the adapter.

Troubleshooting

- On 64-bit machines you must make sure that 32-bit Oracle client components are also installed. This is because Visual Studio 2010 and it's wizards will be running as a 32-bit process requiring access to 32-bit components during development.

Samples

Samples for Microsoft BizTalk Adapter for Oracle Database are categorized into:

- BizTalk Server samples
- WCF service model samples
- WCF channel model samples
- Migration samples

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>. The SQL scripts for creating the tables, packages, etc. used in the samples are also available along with the samples for the Oracle Database adapter.

The following list contains the names and descriptions of the samples for the Oracle Database adapter.

BizTalk Server Samples

Sample Directory Name	Description
Func_RecordTypes	Demonstrates how to use RECORD type parameters and return values in functions and procedures using the Oracle Database adapter.
Func_RefCursor	Demonstrates how to use REF CURSOR parameters in functions and procedures using the Oracle Database adapter.
InvokeFunction	Demonstrates how to invoke a function in Oracle database using the Oracle Database adapter.
InvokeOverloaded Proc	Demonstrates how to invoke with overloaded functions and procedures in Oracle database using the Oracle Database adapter.
Operate_BFILE	Demonstrates how to use Oracle BFILE types in Oracle procedures using the Oracle Database adapter.
Operate_LOB	Demonstrates how to perform ReadLOB and UpdateLOB operations on tables with LOB data types using the Oracle Database adapter.
PollingQuery	Demonstrates how to configure a polling query and receive the results using the Oracle Database adapter.
SelectAccTable	Demonstrates how to perform a select query on an Oracle database table using the Oracle Database adapter.
SqlExec	Demonstrates how to perform parameterized queries using the SQLEXECUTE operation on an Oracle database using the Oracle Database adapter.

WCF Service Model Samples

Sample Directory Name	Description
OracleBfileTypeSM	Demonstrates how to use Oracle BFILE types in basic SQL operations surfaced for Oracle tables and as parameters to Oracle procedures.
OracleOverloadsSM	Demonstrates how to invoke overloaded functions and procedures in a package.
OraclePollingSM	Demonstrates how to configure a polling query and receive the results.

OracleRecordTypesSM	Demonstrates how to use RECORD type parameters and return values in functions and procedures.
OracleRefCursorsSM	Demonstrates how to use REF CURSOR parameters in functions and procedures
OracleTransactedDmlSM	Demonstrates how to perform operations on the Oracle database in a transaction using the WCF service model.

WCF Channel Model Samples

Sample Directory Name	Description
OraclePollingCM	Demonstrates how to configure a polling query and receive the results.
OracleStreamingDemo	Demonstrates how to perform end-to-end streaming of LOB data using the UpdateLOB and table Select operations
OracleTransactedDmlCM	Demonstrates how to perform operations on the Oracle database in a transaction using the WCF channel model.

Migration Samples

Sample Directory Name	Description
Oracle_Migration	Demonstrates how to use a BizTalk project created using the BizTalk ODBC Adapter for Oracle Database (shipped with BizTalk Server) and make it work with the WCF-based Oracle Database adapter (shipped with BizTalk Adapter Pack 2010).

See Also

Concepts

[Development](#)

How Do I?

The topics in this section describe how to perform various operations using the Microsoft BizTalk Adapter for Oracle Database.

In This Section

- [Configure Transaction Isolation Level and Transaction Timeout](#)

See Also

Concepts

[Development](#)

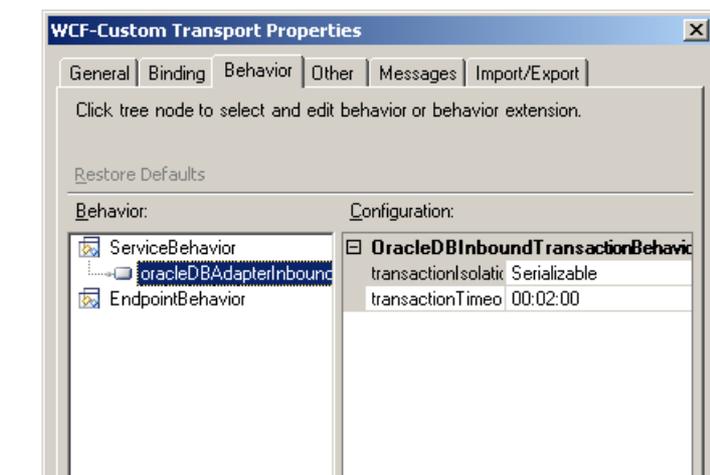
Configure Transaction Isolation Level and Transaction Timeout

While performing inbound operation (Polling) using the Microsoft BizTalk Adapter for Oracle Database with BizTalk Server, you should appropriately configure the transaction isolation level and the transaction timeout values. To do this:

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, and then expand **Applications**.
3. Expand the BizTalk application that you have deployed after generating the metadata using the Oracle Database adapter.
4. Right-click **Receive Ports**, point to **New**, and then click **One-way Receive Port**.
5. In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
6. In the left pane of the **Receive Port Properties** dialog box, click **Receive Locations**, and then click **New** in the right pane to define a new receive location.
7. In the **Receive Location Properties** dialog box, click **WCF-Custom** in the **Type** list.
8. Click **Configure** adjacent to the **Type** list.
9. In the **WCF-Custom Transport Properties** dialog box, click the **Behavior** tab.
10. In the **Behavior** list, right-click **ServiceBehavior**, and click **Add extension**.
11. In the **Select Behavior Extension** dialog box, select **oracleDBAdapterInboundTransactionBehavior**, and click **OK**.
12. In the left pane of the **WCF-Custom Transport Properties**, select the **oracleDBAdapterInboundTransactionBehavior** service under **ServiceBehavior**.
13. In the right pane of the **WCF-Custom Transport Properties**, specify appropriate values for the **transactionIsolationLevel** and **transactionTimeout** parameters. You can select any of the following transaction isolation levels: **Serializable**, **RepeatableRead**, **ReadCommitted**, **ReadUncommitted**, **Snapshot**, **Chaos**, and **Unspecified**. For information about these transaction isolation levels, see the **Members** section at <http://go.microsoft.com/fwlink/?LinkId=126983>.

Important

The Oracle Database adapter supports only the following two transaction isolation levels: ReadCommitted and Serializable.



14. Click **OK** in the **WCF-Custom Transport Properties** dialog box.
15. Click **OK** in the open dialog boxes to save the changes.

Deployment

This section provides information about deploying the Microsoft BizTalk Adapter Pack. It also provides some frequently asked questions and a link to information about troubleshooting common installation issues.

In This Section

- [Installing the BizTalk Adapter Pack](#)
- [Troubleshooting Your Installation](#)
- [Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Installing the BizTalk Adapter Pack

See the BizTalk Adapter Pack installation guide for installation prerequisites and complete instructions for installing the BizTalk Adapter Pack. This document is installed on your computer, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

See Also

Tasks

[Troubleshooting Your Installation](#)

Concepts

[Deployment](#)

Troubleshooting Your Installation

Installation of the Microsoft BizTalk Adapter Pack copies the product binaries on a computer and registers the bindings for each adapter. For information about troubleshooting BizTalk Adapter Pack installation issues, see [Troubleshooting Installation Issues](#).

See Also

Tasks

[Troubleshooting Your Installation](#)

Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database

The BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server differs from Microsoft BizTalk Adapter for Oracle Database that is shipped with BizTalk Adapter Pack in many aspects, including:

- The design-time experience of creating a BizTalk project.
- The metadata retrieval experience.
- Schemas.
- Data type mappings.
- The operations that can be performed using the adapter.
- Physical port configuration in the BizTalk Server Administration console.

This section discusses the differences between these two adapters.

Note

The Oracle Database adapter that is shipped with BizTalk Adapter Pack is a WCF-based adapter and can be used with BizTalk Server, the WCF service model, or the WCF channel model. The BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server, however, can be used only with BizTalk Server.

Therefore, this section talks about version differences in terms of using the adapter with BizTalk Server 2010. For information about how to use the adapter with the WCF service model or WCF channel model, see [Overview of BizTalk Adapter for Oracle Database](#).

In This Section

- [Differences in the Design-time Experience](#)
- [Differences in Metadata Retrieval Experience](#)
- [Differences in the Schema File Name and Namespace](#)
- [Differences in Data Type Mapping](#)
- [Differences in Operations Surfaced for the Oracle Database](#)
- [Differences in Physical Port Configuration in the BizTalk Server Administration Console](#)
- [Migration Tutorials and Sample](#)

See Also

Concepts

[Deployment](#)

Differences in the Design-time Experience

The design-time experience for using the adapters includes creating a BizTalk project in Visual Studio and then generating metadata for the operations you want to perform on the Oracle database using the adapter. This section discusses how the design-time experience differs between the two adapter versions.

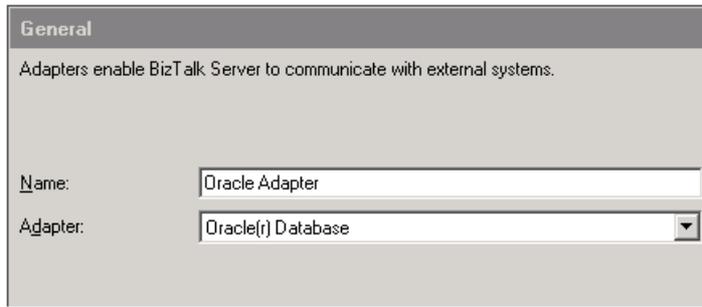
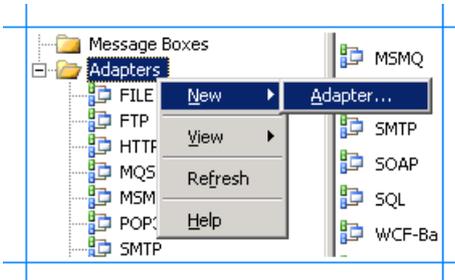
Important

In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as “vPrev Oracle Database adapter.”

Design-time Experience for the vPrev Oracle Database Adapter

The design-time experience using the vPrev Oracle Database adapter involves the following steps:

1. **Add the Oracle Database adapter to the BizTalk Server Administration console.** Installing the vPrev Oracle Database adapter does not add the adapter to the console. You must add the vPrev Oracle Database adapter in the BizTalk Server Administration console under BizTalk Server Administration > BizTalk Group > Platform Settings > Adapters.

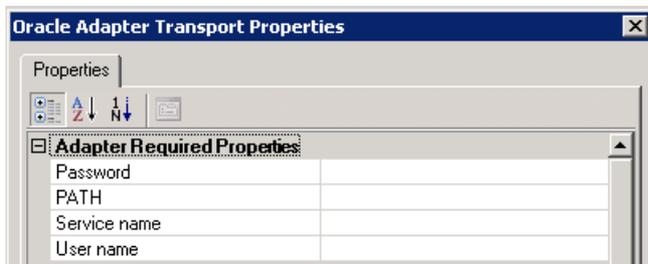


2. **Add a physical send port.** To retrieve metadata from an Oracle database, the vPrev Oracle Database adapter requires you to configure a physical send port in BizTalk Server Administration console, before you create a BizTalk project.

You can configure a send port under BizTalk Server Administration > BizTalk Group > BizTalk Applications > [Your Application] > Send Ports. For the send port, you must specify the connection properties to connect to the Oracle database.

The key connection properties to connect to an Oracle database using the vPrev Oracle Database adapter are:

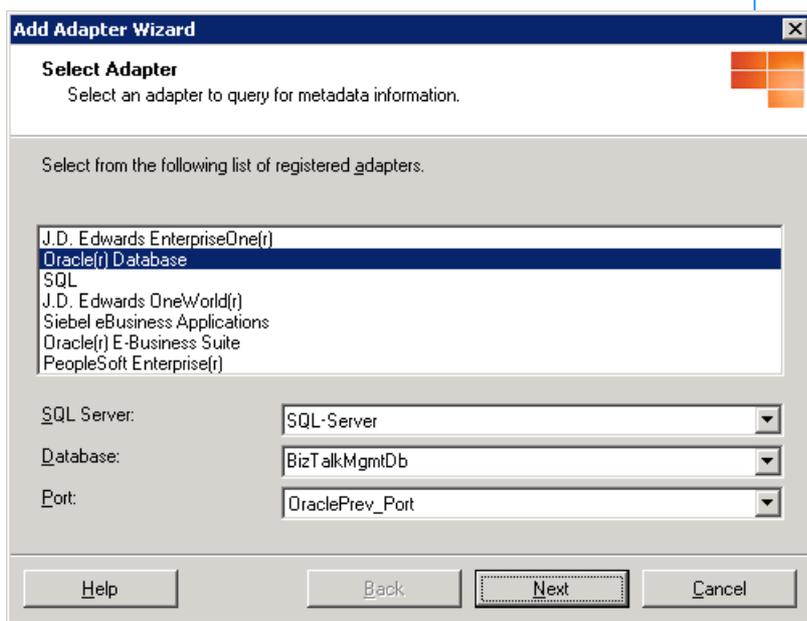
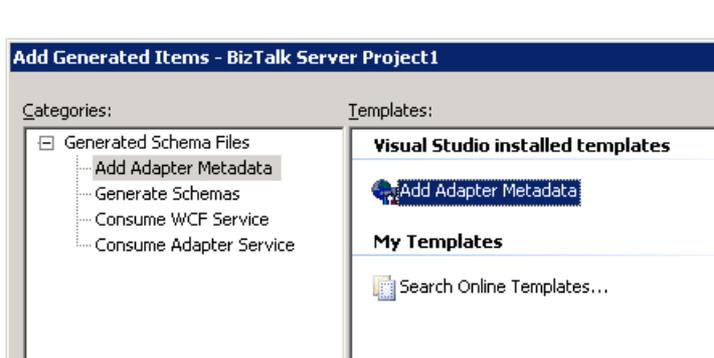
- **Without Enterprise SSO:** Password, PATH, service name, user name, Use SSO (set to No).
- **With Enterprise SSO:** ESSO affiliate application, service name, Use SSO (set to Yes).



The service name in the connection properties refers to the system Data Source Name (DSN). The DSN name is configured under Data Sources (ODBC) by going to Start > Control Panel > Administrative Tools > Data Sources (ODBC).

The PATH is the path of the Oracle "bin" directory.

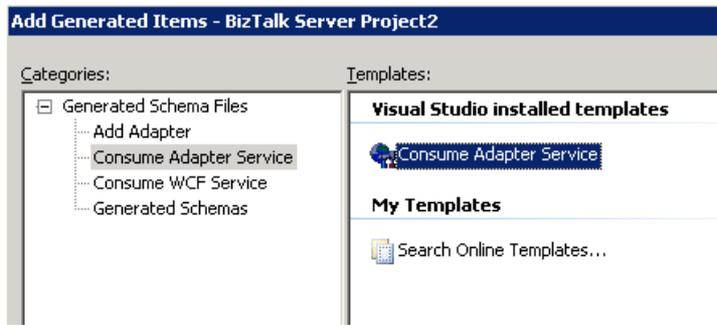
3. **Generate metadata.** Create a BizTalk project in Visual Studio. Right-click the BizTalk project, and then select Add > Add Generated Items > Add Adapter Metadata.



Design-time Experience for the WCF-based Adapter

The design-time experience using the WCF-based Oracle Database adapter involves the following steps:

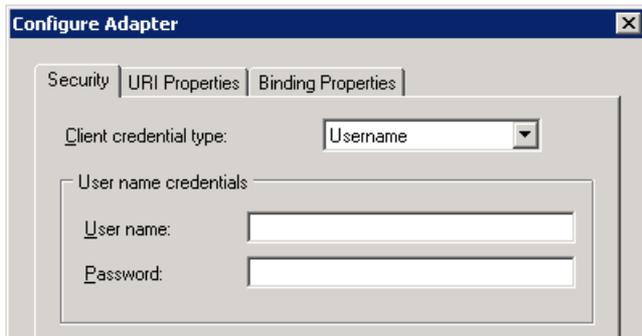
1. **Create a BizTalk project.** Create a BizTalk project in Visual Studio. Right-click the BizTalk project, and then select Add > Add Generated Items > Consume Adapter Service.



Note

Note that WCF-based Oracle Database adapter does not require you to configure a physical BizTalk port for the design-time experience.

2. **Connect to the Oracle database.** In the Consume Adapter Service BizTalk Project Add-in, select the binding for Oracle (oracleDBBinding) and click **Configure** to start the **Configure Adapter** dialog box. You must specify the different connection properties in this dialog box to connect to the Oracle database.



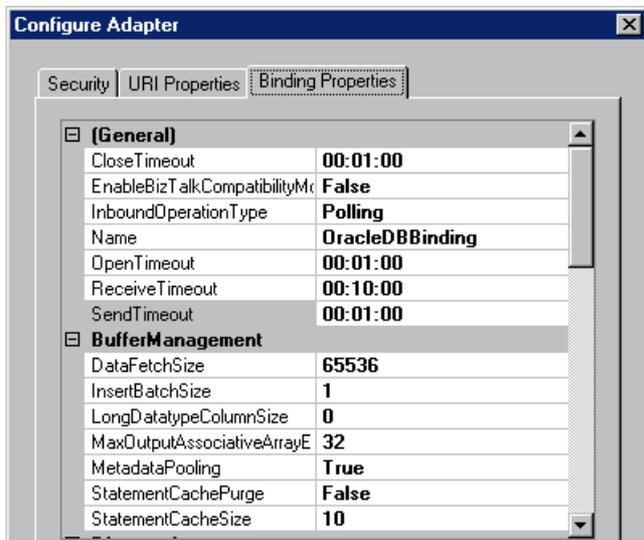
The key connection properties to connect to an Oracle database using the WCF-based Oracle Database adapter are:

- **User name.** Oracle user names are stored in all CAPS by default in the database. For example, user scott is internally stored as SCOTT. Because the adapter preserves the case in which you specify the password, you must provide the user name in all CAPS.
- **Password** for the user. For release 10g and earlier, passwords on the Oracle system are not case-sensitive. The Oracle Database adapter preserves the case of the value that you enter for the password when it opens a connection on the Oracle database. Oracle introduced case-sensitive passwords starting with Oracle 11g.
- **Oracle data source name** that can be configured either in the Oracle client's tnsnames.ora file or in a directory server.

If you are using the WCF-based Oracle Database adapter to receive polling-based data-changed messages from the Oracle database, the **PollingId** parameter also becomes relevant. The value you specify for **PollingId** is appended to the POLLINGSTMT operation surfaced by the adapter. This enables BizTalk users to have more than one receive location for a BizTalk application for the same Oracle data source. For example, if you want to poll data from two different tables as part of the same BizTalk application, you could do this by creating two POLLINGSTMT schemas. You could use the connection URI `oracledb://MYDATASOURCE?PollingId=1` to create the first schema and the connection URI `oracledb://MYDATASOURCE?PollingId=2` for the second schema.

For more information about the connection properties, see [Establishing a Connection to the Oracle Database](#).

3. **Specify binding properties.** The WCF-based Oracle Database adapter is essentially a custom WCF binding that includes different binding properties. These binding properties govern the behavior of the adapter. These properties can be set both at design time (when generating the metadata) and at run time (when sending messages to the Oracle database).



If you want to enable polling, the key binding property that must be set at design time is **PollingStatement**. The adapter uses the value for the **PollingStatement** property to generate the metadata of a special inbound operation called the "POLLINGSTMT". You can either specify a SELECT query or stored procedure, function, or procedure or function within a package to poll the Oracle database.

For more information about how the Oracle Database adapter supports polling, see [Receiving Polling-based Data-changed Messages](#).

Key Differences in Design-time Experience

The following table lists the key differences in the design-time experience for the two versions of the adapter.

vPrev Oracle Database adapter	WCF-based Oracle Database adapter
Add the adapter to the BizTalk Server Administration console under Platform Settings > Adapters .	The WCF-based Oracle Database adapter is a WCF custom binding and is configured as specific binding type of the WCF-Custom adapter. The BizTalk Server Administration console already contains the WCF-Custom adapter under Platform Settings > Adapters .
Create a physical send port to connect to the Oracle database at design time.	There is no need to configure a send port.
Use the Add Adapter Metadata option to connect to the Oracle database and retrieve metadata.	Use the Consume Adapter Service option to connect to the Oracle database and retrieve metadata.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Differences in Metadata Retrieval Experience

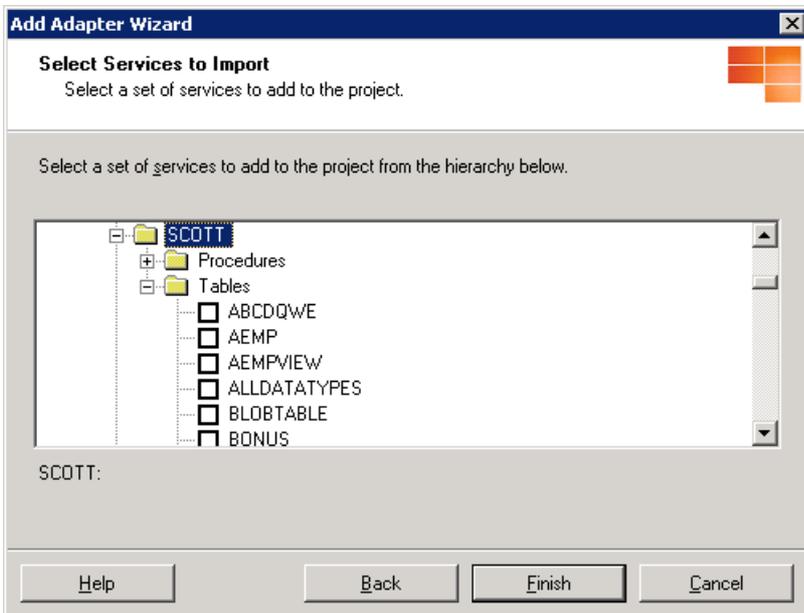
Once you have connected to the Oracle database from the BizTalk project in Visual Studio, you can generate metadata to perform operations on the Oracle database, such as an Insert operation on the SCOTT.EMP table. The steps involved in generating the metadata in the BizTalk ODBC Adapter for Oracle Database differ from those in the WCF-based Oracle Database adapter. This section discusses the differences.

Important

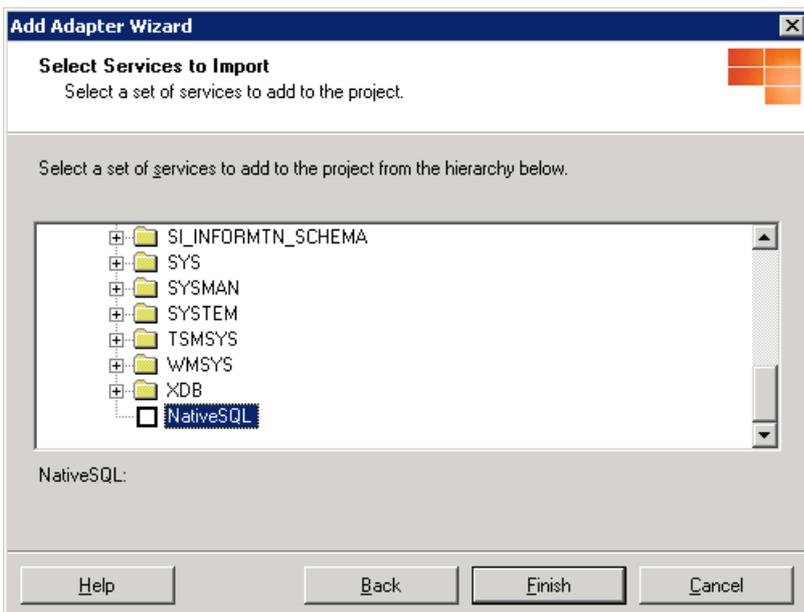
In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as "vPrev Oracle Database adapter."

Metadata Retrieval Experience for the vPrev Oracle Database Adapter

- Metadata categories.** The vPrev Oracle Database adapter surfaces two categories under an Oracle schema: **Table** and **Procedure**. All the tables belonging to a particular schema are surfaced under the **Table** node. Similarly, all the packages are surfaced under the **Procedure** node. However, no operations (Insert, Update, Select, Delete) are surfaced for the tables. Similarly, procedures and functions constituting the package are not surfaced under the **Procedure** node.



- Special operations.** The vPrev Oracle Database adapter surfaces a special node called NativeSQL. This operation is used to execute SQL statements in the Oracle database.

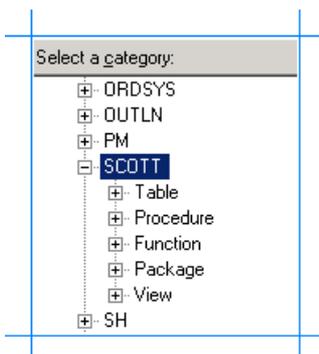


- Searching for Oracle database artifacts.** The vPrev Oracle Database adapter does not support searching for Oracle

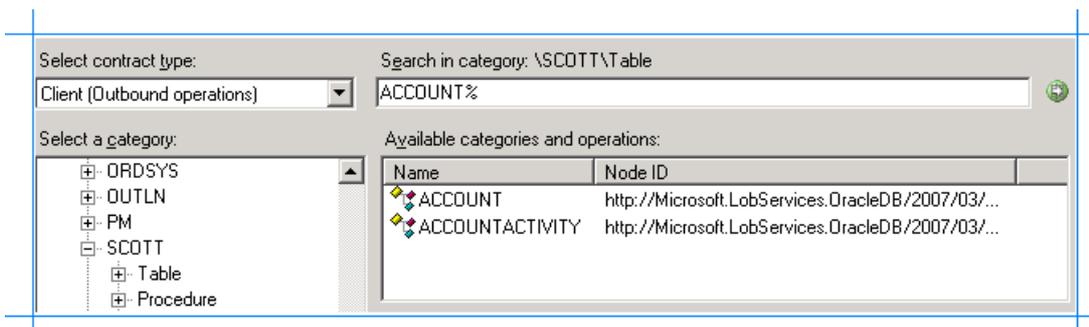
artifacts. You need to browse through the entire tree exposed by the adapter to find the artifact you are looking for.

Metadata Retrieval Experience for the WCF-based Adapter

- **Contract type.** Based on whether you will be performing outbound operations (messages sent from a client to the Oracle database) or inbound operations (messages sent from Oracle database to a client) using the adapter, you can specify the contract type.
 - For outbound operations, the contract type is Client. This contract surfaces operations whose messages flow from BizTalk to the adapter via a send port.
 - For inbound operations, the contract type is Service. This contract surfaces operations whose messages flow from the adapter to BizTalk via a receive port.
- **Metadata categories.** The WCF-based Oracle Database adapter surfaces the **Table**, **Procedure**, **Function**, **Package**, and **View** categories. The **Table** node surfaces all the tables for a particular schema. In addition to this, the adapter also surfaces the different operations (Insert, Update, Select, Delete) that can be performed on the table. The standalone functions and procedures are surfaced under the **Function** and **Procedure** nodes, respectively. Procedures and functions constituting a package are surfaced under the **Package** node. Each node also surfaces the synonyms for the respective artifacts. For example, the **Table** node under a schema will display all the synonyms for tables along with the database tables in a schema, the **View** node under a schema will display all the synonyms for views along with the database views in a schema, and so on.



- **Special operations.** The WCF-based Oracle Database adapter surfaces a special node called SQLEXECUTE under the root node. This operation is used to execute SQL statements in the Oracle database. For inbound operations, the Oracle Database adapter surfaces two special nodes called POLLINGSTMT and Notification. POLLINGSTMT is used to receive polling-based data change messages from the Oracle database. Notification is used to register a SELECT statement as the notification query on the database, and receive notifications from the database as and when the result set of the SELECT statement changes.
- **Searching for Oracle database artifacts.** With the WCF-based Oracle Database adapter you can specify a search expression after selecting a relevant category node. For example, to search for a specific table, select the **Table** node and specify a search expression similar to an Oracle query search expression. Note that the search expressions are case-sensitive. For more information about searching for Oracle database artifacts using the WCF-based Oracle Database adapter, see [How Does the Adapter Surface Oracle Metadata?](#)



- **BizTalk port binding file.** Generating schema for the WCF-based Oracle Database adapter using the Consume Adapter Service BizTalk Project Add-in also creates a BizTalk port binding file. You can import this file into a BizTalk application to

configure a WCF-Custom port. The file essentially contains the physical port configuration including the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. This makes it significantly easier to create physical WCF-Custom ports for the project. For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Key Differences in Metadata Retrieval Experience

The following table lists the key differences in the metadata retrieval experience for the two versions of the adapter.

vPrev Oracle Database adapter	WCF-based Oracle Database adapter
Only Table and Procedure categories are surfaced.	Table, Procedure, Function, Package, and View categories are surfaced.
No operations are surfaced under the Table node.	Insert, Select, Update, Delete operations are surfaced under the Table node.
Special NativeSQL operation is surfaced.	Special SQLEXECUTE, POLLINGSTMT, and Notification operations are surfaced.
Search is not supported.	Search is supported for Oracle database artifacts using a search expression similar to an Oracle query search expression.
There is no differentiation between inbound and outbound operations.	Inbound operations are surfaced under the Service (Inbound Operations) category, whereas outbound operations are surfaced under the Client (Outbound Operations) category.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Differences in the Schema File Name and Namespace

The BizTalk ODBC Adapter for Oracle Database generates metadata for Oracle artifacts differently than the WCF-based Oracle Database adapter. This section discusses these differences including:

- File naming convention.
- Target namespace.

Important

In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as “vPrev Oracle Database adapter.”

Schema File Name and Namespace for the vPrev Oracle Database Adapter

- **File naming convention.** The vPrev Oracle Database adapter generates schema files with names containing the name of the Oracle database artifact. For example, generating the metadata for the EMPLOYEE table creates a file with the name EMPLOYEEService_EMP_x5d.xsd.
- **Target namespace of the schema.** The following table lists the target namespace of the schemas generated by the previous version of the adapter.

Oracle database artifact	Target namespace
Table	http://schemas.microsoft.com/[OracleDb://{DSN_Name}/{Schema}/Tables/{Table_Name}]
Procedure	http://schemas.microsoft.com/[OracleDb://{DSN_Name}/{Schema}/Procedures/{Package_Name}]
NativeSQL	http://schemas.microsoft.com/[OracleDb://{DSN_Name}/NativeSQL]

In the preceding table, the text in parentheses { } are placeholders to present the convention for the namespace. The actual namespace contains the appropriate values.

Schema File Name and Namespace for the WCF-based Adapter

- **File naming convention.** By default, the WCF-based Oracle Database adapter generates schema files with names containing the name of the binding and the object on which the operation is performed. For example, generating the metadata for the EMPLOYEE table will create a binding file with the name as OracleDBBindingSCOTT.Table.Employee.xsd. However, the Consume Adapter Service BizTalk Project Add-in provides the option of specifying a custom file name while generating the schema. For more information about how to retrieve the schema, see [Retrieving Metadata for Oracle Operations in Visual Studio](#).
- **Target namespace of the schema.** The following table lists the target namespace of the schemas generated by the WCF-based Oracle Database adapter.

Oracle database artifact	Target namespace
Table	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Table/{Table_Name}
Procedure	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Procedure

Procedure Overloads	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Procedure/{Procedure_Name}/overload{count}
Function	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Function
Function Overloads	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Function/{Function_Name}/overload{count}
Package	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Package/{Package_Name}
Package Overloads	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/Package/{Package_Name}/{Procedure_Name/Function_Name}/overload{count}
View	http://Microsoft.LobServices.OracleDB/2007/03/{Schema}/View/{View_Name}
SQLEXECUTE	http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE
POLLINGSTMT	http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMT{PollingId}
Notification	http://Microsoft.LobServices.OracleDB/2007/03/Notification

In the preceding table, the text in parentheses { } are placeholders to present the convention for the namespace. The actual namespace contains the appropriate values.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Differences in Data Type Mapping

The BizTalk ODBC Adapter for Oracle Database and the WCF-based Oracle Database adapter differ in how the Oracle data type maps to the XSD data type. This section lists the data type mapping in both versions of the adapter.

Important

In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as "vPrev Oracle Database adapter."

Data Type Mapping

The following table lists the mapping between the Oracle data types and XSD data types for both the adapter versions. For more information about data type mapping for WCF-based Oracle Database adapter, see [Basic Oracle Data Types](#).

In the WCF-based Oracle Database adapter, how an Oracle data type is mapped to an XSD type depends on whether the **EnableSafeTyping** binding property is set to **True** while generating the schema. For more information about the binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Oracle data type	XSD mapping for vPrev Oracle Database adapter	XSD mapping for WCF-based Oracle Database adapter
BFile	xsd:base64Binary	<ul style="list-style-type: none"> input: xsd:string output: xsd:base64Binary
Blob	xsd:base64Binary	xsd:base64Binary
Clob	xsd:string	xsd:string
NClob	xsd:string	xsd:string
Raw (any precision)	xsd:base64Binary	xsd:base64Binary
LongRaw	xsd:base64Binary	xsd:base64Binary
Long	xsd:string	xsd:string
Number with precision > 28 *	xsd:string with number pattern	xsd:string
Number with precision < = 28 *	xsd:string with number pattern	xsd:decimal
Binary Float with precision > 49 (decimal precision 15) *	xsd:string with number pattern	xsd:string
Binary Float with precision < = 23 (decimal precision 7) *	xsd:string with number pattern	xsd:float
Binary Float with precision > 23 and < = 49 *	xsd:string with number pattern	xsd:double

Char	xsd:string	xsd:string
NChar	xsd:string	xsd:string
Varchar2	xsd:string	xsd:string
NVarchar2	xsd:string	xsd:string
Date	xsd:dateTime	<ul style="list-style-type: none"> • xsd:dateTime if EnableSafeTyping = False • xsd:string if EnableSafeTyping = True; the value should be expressed in Oracle NLS_DATE_FORMAT.
TimeStamp with precision <= 7	xsd:dateTime	<ul style="list-style-type: none"> • xsd:dateTime if EnableSafeTyping = False • xsd:string if EnableSafeTyping = True; the value should be expressed in Oracle NLS_TIMESTAMP_FORMAT.
Timestamp with precision > 7	xsd:dateTime	xsd:string; the value should be expressed in Oracle NLS_TIMESTAMP_FORMAT.
TimeStampLTZ	xsd:base64Binary	xsd:string; the value should be expressed in NLS_TIMESTAMP_TZ_FORMAT.
TimeStampTZ	xsd:base64Binary	<ul style="list-style-type: none"> • xsd:string if outside an UDT; the value should be expressed in NLS_TIMESTAMP_TZ_FORMAT. • xsd:dateTime if inside an UDT
IntervalYM	xsd:base64Binary	<ul style="list-style-type: none"> • xsd:string if outside an UDT; the value should be expressed in Oracle native format: Year-Month; for example, "1-2" (1 year and 2 months). • xsd:long if inside an UDT
IntervalDS	xsd:base64Binary	<ul style="list-style-type: none"> • xsd:string if outside an UDT; the value should be expressed in Oracle native format: Day HH:MI:SSxFF; for example, "5 15:30:12.99". • xsd:duration if inside an UDT

* If safe typing is enabled (**EnableSafeTyping** = True), these Oracle numeric data types inside DataSets and weakly-typed REF CURSORS in the WCF-based Oracle Database adapter are always exposed as strings.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Differences in Operations Surfaced for the Oracle Database

The two versions of the adapter differ in the operations they expose for the Oracle database. This section lists the differences in:

- Operations surfaced for tables.
- Operations on functions and procedures.
- Special operations.

For more information about the operations supported by the WCF-based Oracle Database adapter, see [What Operations Can be Performed Using the Adapter?](#)

Important

In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as "vPrev Oracle Database adapter."

Operations Surfaced for Tables

The following table lists the operations surfaced for tables by both versions of the adapter.

Operation for the vPrev Oracle Database adapter	Description	Operation for the WCF-based Oracle Database adapter	Description
Insert	Inserts one or more records into a table.	Insert	<ul style="list-style-type: none"> • Inserts one or more records into a table. • Performs a bulk insert into a table using a SELECT query and column names.
Query	Selects rows based on a filter (where clause) and max rows.	Select	Selects rows based on column and row filter.
Update	Updates one or more records based on a filter and record field values to be updated.	Update	Updates one or more records based on a filter and record field values to be updated.
Remove	Updates one or more records based on a filter.	Delete	Updates one or more records based on a filter.
TableChangeEvent	Provides all rows and all columns in a table at regular intervals of time (polling). Deletes rows after the poll.	POLLINGSTMT	Provides rows in a table with row and column filtering at regular intervals of time (polling) as a strongly-typed result set. Executes a post poll statement (PL/SQL block) provided by the user after the poll.
		ReadLob	Reads the value of a specific LOB column in a table.
		UpdateLob	Updates the value of a specific LOB column in a table.

Operations on Procedures and Functions

- For procedures, the vPrev Oracle Database adapter surfaces the procedures and functions constituting packages as operations. However, the WCF-based Oracle Database adapter surfaces standalone stored procedures and functions, in addition to stored procedures and functions constituting packages as operations. For the inbound operation, a list of procedures, functions, and packages in the schema are exposed as operations for polling.

- While the vPrev Oracle Database adapter does not handle procedure and function overloads, the WCF-based Oracle Database adapter handles overloads by generating separate target namespaces. Typically, namespaces are suffixed with overload1, overload2, and so on. For example:

http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/OVERLOAD/GET_RECORD/overload1

Special Operations Surfaced by the Adapters

The following table lists the differences in the special operations surfaced by both versions of the adapter.

Operation for the vPrev Oracle Database adapter	Description	Operation for WCF-based Oracle Database adapter	Description
NativeSQL	Executes a generic SQL statement. Provides ability to specify IN parameter values.	SQLEXECUTE	Executes a generic SQL statement. Provides ability to specify IN parameter values.
SQLEvent	Provides rows in a table with row and column filtering at regular intervals of time (polling) as a generic result set containing column metadata and row data. Executes a post poll statement provided by the user after the poll.	POLLINGSTMT	Obtains inbound data from the Oracle database based on a query polling mechanism supported by the adapter.
		Notification	Registers a SELECT statement as the notification query on the database, and the database sends a notification to the adapter client as and when the result set of the SELECT statement changes.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Differences in Physical Port Configuration in the BizTalk Server Administration Console

To use the BizTalk ODBC Adapter for Oracle Database or the WCF-based Oracle Database adapter with BizTalk Server, you need to configure physical ports in the BizTalk Server Administration console. This section lists the differences in configuring a physical port for both versions of the adapter.

◆ Important

In this topic, for the sake of brevity, the BizTalk ODBC Adapter for Oracle Database that is shipped with Microsoft BizTalk Server will be referred to as "vPrev Oracle Database adapter."

Configuring a Physical Port for the vPrev Oracle Database Adapter

You can configure a send port to send messages to the Oracle database using the vPrev Oracle Database adapter, or a receive port to receive messages from the Oracle database using the vPrev Oracle Database adapter. The configuration properties displayed for the send and receive ports are different. For example, while configuring a receive port for the previous version of the Oracle database adapter, the properties related to polling statement or table change event are available. However, these properties are not available while configuring a send port.

Configuring a Physical Port for the WCF-based Oracle Database Adapter

For the WCF-based Oracle Database adapter, you do not necessarily have to create a port manually. You have the option of configuring a physical port automatically by importing a port binding file that is generated while you generate metadata for any operation using the Consume Adapter Service BizTalk Project Add-in. The file essentially contains the physical port configuration including the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. For more information, see [Configuring a Physical Port Binding by Using a Port Binding File](#).

Several other considerations that make a physical port configuration for WCF-based Oracle Database adapter different from the previous version of the Oracle database adapter are:

- The port type for WCF-based Oracle Database adapter is either "WCF-Custom" or "WCF-OracleDB". So, with respect to BizTalk Server, this port uses custom WCF configurations and primarily, a custom WCF binding. Because WCF-based Oracle Database adapter is essentially a custom WCF binding, it can be used in BizTalk via the WCF-Custom or WCF-OracleDB port.
- A BizTalk WCF-Custom or WCF-Oracle DB send port requires an "Action" to be specified in the port. This is the WCF action that is sent on the message by BizTalk Server before it gets to the binding. The WCF action conveys to the adapter:
 - The operation the adapter client wants to perform.
 - The Oracle database artifact on which the user wants to perform the operation.

Adapter clients can specify this action in various ways. For more information on specifying an action, see [Specifying SOAP Action](#).

- BizTalk WCF-Custom or WCF-Oracle DB send or receive ports includes binding information that, in turn, includes a binding type and binding properties for that binding type. This is a key configuration that ties the WCF-Custom port or WCF-Oracle DB to a WCF-based Oracle Database adapter. For the WCF-based Oracle Database adapter, binding type is **oracleDBBinding**.
- The binding configuration properties that govern how an adapter behaves while generating metadata or while sending messages to the Oracle database. Unlike the previous vPrev Oracle Database adapter, the available binding properties do not change depending on whether you are configuring a send or a receive port. However, adapter users must be aware of which binding property to set for a specific operation performed by the adapter. For more information about the binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).
- The configurations under the **Messages** tab in the **Transport Properties** dialog box of the port enable adapter users to

alter the message that is sent or received using the WCF-Custom or WCF-OracleDB port. While sending messages (from BizTalk Server to the adapter), users can choose to send the message body or use a template to transform the XML message to a different message structure before sending it to the adapter (the WCF custom binding). While receiving messages (from the adapter to BizTalk), users can choose to get the entire SOAP envelope, the body contents of the SOAP message, or a specific node value in the incoming message referenced by an XPath query expression.

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Migration Tutorials and Sample

The Microsoft BizTalk Adapter Pack includes migration tutorials that demonstrate how to migrate BizTalk projects created using the BizTalk ODBC Adapter for Oracle Database to use the WCF-based Oracle Database adapter. For more information, see [Tutorial: Migrating BizTalk Projects](#).

The BizTalk Adapter Pack also includes a sample that demonstrates how to modify a BizTalk project created using the BizTalk ODBC Adapter for Oracle Database to send and receive messages using the WCF-based Oracle Database adapter. For more information, see [Samples](#).

See Also

Other Resources

[Migrating BizTalk Projects Created Using the BizTalk ODBC Adapter for Oracle Database](#)

Troubleshooting

The BizTalk Adapter Pack uses or depends on several Microsoft technologies, including but not limited to the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, Microsoft Visual Studio, and the Microsoft .NET Framework 4 or .NET Framework 3.5 SP1. The BizTalk Adapter Pack is built on top of the WCF LOB Adapter SDK, which in turn requires the .NET Framework 4 or .NET Framework 3.5 SP1. The adapters can be consumed either by writing applications using Visual Studio or by creating BizTalk applications. For issues related to each of these technologies and products, see the respective documentation.

This section provides information about troubleshooting the BizTalk Adapter Pack, including:

- Enabling tracing to diagnose issues with the adapters.
- Handling installation and operational issues that you might encounter when working with the adapters, including probable cause, and a resolution.
- Using performance counters to gauge adapter performance.
- Handling exceptions and errors, including probable cause, and a resolution.

In This Section

- [Diagnostic Tracing and Message Logging](#)
- [Troubleshooting Installation Issues](#)
- [Troubleshooting Operational Issues](#)
- [Troubleshooting Performance Issues](#)
- [Using Performance Counters](#)
- [Exceptions and Error Handling](#)

Diagnostic Tracing and Message Logging

Diagnostic tracing helps to effectively diagnose problems that you might encounter when using the adapters. This topic provides information about the following two types of tracing supported with Microsoft BizTalk Adapter for Oracle Database:

- WCF tracing between the adapter client and the adapter
- WCF tracing within the adapter

WCF Tracing Between the Adapter Client and the Adapter

Adapter clients can enable WCF tracing to trace issues between the adapter client and the adapter. WCF tracing is used to trace the input XML that comes from the adapter client by using the WCF service model, and is useful in diagnosing serialization issues. WCF tracing is not used for the WCF channel model or for output messages from the adapter to the adapter client. You can activate WCF tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in `<installation drive>\Program Files\Microsoft Visual Studio <version>\Common7\IDE`.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically `BTSNTSvc.exe.config`. For BizTalk Server 2010, this file is available typically under `<installation drive>\Program Files\Microsoft BizTalk Server 2010`.
 - For a WCF service model .NET application, you must add the excerpt to the `app.config` file of your project.

To enable WCF tracing, add the following excerpt within the `<configuration>` tag.

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.ServiceModel.MessageLogging"
      switchValue="Verbose, ActivityTracing">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.Runtime.Serialization" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\WCFTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>
<system.serviceModel>
  <diagnostics>
```

```

    <messageLogging
      logEntireMessage="true"
      logMalformedMessages="false"
      logMessagesAtServiceLevel="true"
      logMessagesAtTransportLevel="false"/>
  </diagnostics>
</system.serviceModel>

```

This saves the WCF traces to C:\log\WCFTrace.svclog. For more information about WCF tracing, see "Tracing" at <http://go.microsoft.com/fwlink/?LinkId=91242>.

◆ Important

Make sure you mitigate potential security threats of exposing sensitive business data by enabling tracing. For recommendations see [Best Practices](#).

WCF Tracing Within the Adapter

The adapters log different categories of useful information to the trace file such as errors, warnings, and information messages. Such information is useful in understanding the process flow within the adapter and diagnosing issues with the adapter. You can activate the WCF LOB Adapter SDK and adapter tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in <installation drive>\Program Files\Microsoft Visual Studio <version>\Common7\IDE.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically BTSNTSvc.exe.config. For BizTalk Server 2010, this file is available typically under <installation drive>\Program Files\Microsoft BizTalk Server 2010.
 - For a WCF service model .NET application, you must add the excerpt to the app.config file of your project.

To enable WCF LOB Adapter SDK and adapter tracing, add the following excerpt within the <configuration> tag:

```

<system.diagnostics>
  <sources>
    <source name="Microsoft.ServiceModel.Channels" switchValue="Error">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name=" Microsoft.Adapters.OracleDB" switchValue="Information">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\AdapterTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>

```

This saves the WCF traces to C:\log\AdapterTrace.svclog.

Viewing the Traces

You can use the Windows Communication Foundation (WCF) Service Trace Viewer tool to view the traces. For more information about the tool, see "Using Service Trace Viewer for Viewing Correlated Traces and Troubles" at <http://go.microsoft.com/fwlink/?LinkId=91243>.

Configuring Tracking for BizTalk Applications

The BizTalk Server Administration console lets you configure various tracking options for items such as send ports and receive ports. The tracking configuration settings enable you to track inbound and outbound event data, message properties, message bodies, and orchestrations. For more information about configuring tracking for BizTalk applications, see the "Configuring Tracking Using the BizTalk Server Administration Console" at <http://go.microsoft.com/fwlink/?LinkId=106851>.

You can also use Health and Activity Tracking (HAT) to view historical or tracked data. For more information, see "Viewing Historical or Tracked Data Using HAT" at <http://go.microsoft.com/fwlink/?LinkId=106852>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Installation Issues

This topic was last updated on: June 05, 2009

Installation of the Microsoft BizTalk Adapter Pack copies the product binaries on the computer and registers the bindings for each adapter. This section discusses using troubleshooting techniques to resolve installation errors.

Logging Messages for Setup Actions

The BizTalk Adapter Pack setup program performs the standard task of installing the BizTalk Adapter Pack. Additionally, the setup also performs certain custom actions such as registering the adapter bindings. You can log messages for both the standard as well as custom actions that the setup performs.

- The BizTalk Adapter Pack setup installs the adapter-specific files using an MSI. Therefore, the logging for the setup is the standard MSI logging. For more information about logging for an MSI, see <http://go.microsoft.com/fwlink/?LinkId=196953>.
- All logs for the custom actions that the setup program performs are available at %TEMP%\adaptersetup.log. If the tracing to the log file fails, the traces are also available in the event log.

Known Issues

The following are the most common errors you might encounter when installing the BizTalk Adapter Pack, along with their probable cause and resolution.

- [Setup fails to register adapter bindings](#)
- [Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation](#)
- [Invalid binding error while configuring Oracle database adapter ports in BizTalk Server Administration Console on a 64-bit installation](#)

Setup fails to register adapter bindings

Problem

The Microsoft BizTalk Adapter Pack setup wizard fails to register the adapter bindings, but proceeds with the adapter installation.

Cause

This might result due to problems with Windows Communication Foundation (WCF) installation, Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK installation, or the machine.config file being corrupt. The adapter bindings are written to the machine.config file.

Resolution

You should manually register the Oracle Database adapter binding.

To register the adapter binding

1. Navigate to the machine.config file on the computer. For example, on a 32-bit platform, the machine.config is available under <system drive>:\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG.

In this path, <version> is the version of the .NET Framework. For example, for Microsoft .NET Framework 3.5 SP1, the version will be v2.0.50727.
2. Open the file by using a text editor.
3. To register the Oracle Database adapter binding:
 - a. Search for the element "system.serviceModel" and add the following under it:

```
<client>  
  <endpoint binding="oracleDBBinding" contract="IMetadataExchange" name="oracleDb" />  
</client>
```

- b. Search for the element "bindingElementExtensions" under system.serviceModel\extensions.
- c. Look for the missing Oracle Database adapter binding. Add the following section under the "bindingElementExtensions" node.

For Oracle Database adapter, add:

```
<add name="oracleDBAdapter" type="Microsoft.Adapters.OracleDB.OracleDBAdapterExtensionElement,Microsoft.Adapters.OracleDB, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

d. Search for the element "bindingExtensions" under system.serviceModel\extensions.

e. Look for the missing Oracle Database adapter binding. Add the following section under the "bindingExtensions" node.

For Oracle Database adapter, add:

```
<add name="oracleDBBinding" type="Microsoft.Adapters.OracleDB.OracleDBAdapterBindingSection,Microsoft.Adapters.OracleDB, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

Note

For information about how to determine the public key and the version, see [Determining the Public Key and Version](#).

4. Save and close the machine.config file.

Determining the Public Key and Version

Perform the following steps to determine the public key for Oracle Database adapter.

To determine the public key

1. Navigate to the Windows directory, typically C:\WINDOWS\assembly.
2. Right-click the DLL for which you want the public key and the version, and then select **Properties**. The following table lists the name of the DLL for Oracle Database adapter.

Adapter	Name of the DLL
Oracle Database adapter	Microsoft.Adapters.OracleDB

3. On the **General** tab, the value against the **Public Key Token** label specifies the public key for the DLL. Similarly, value against the **Version** label specifies the version number for the DLL.
4. Copy the public key, and then click **Cancel**.

Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation

Problem

Using the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in from Visual Studio on a 64-bit computer running 64-bit version of the BizTalk Adapter Pack results in the following error:

No valid adapters are installed on this machine

Cause

The Oracle Database adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, Visual Studio runs as a 32-bit process and hence when you launch the Consume Adapter Service Add-in from Visual Studio, the plug-in checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

- Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

- Install both the 32-bit and 64-bit versions of the Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see "Oracle Data Provider for .NET FAQ" at <http://go.microsoft.com/fwlink/?LinkId=92834>.

Invalid binding error while configuring Oracle database adapter ports in BizTalk Server Administration Console on a 64-bit installation**Problem**

When you try to configure a port for the adapter in BizTalk Server Administration console, you get the following error:

```
"Unable to create binding configuration element for editing. Check the values of the BindingType and BindingConfiguration properties.
(Microsoft.Biztalk.Adapter.Wcf.Converters.CreateBindingException) Unable to get binding type for binding extension "oracleDBBinding".
Verify the binding extension is registered in machine.config."
```

Cause

The Oracle Database adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, BizTalk Server Administration console runs as a 32-bit process and hence when you configure a port for the adapter, it checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

- Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

- Install both the 32-bit and 64-bit versions of the Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see "Oracle Data Provider for .NET FAQ" at <http://go.microsoft.com/fwlink/?LinkId=92834>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Operational Issues

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve operational errors that you might encounter when using Microsoft BizTalk Adapter for Oracle Database.

Enabling Tracing

For information about tracing support in the Oracle Database adapter, see [Diagnostic Tracing and Message Logging](#).

Known Issues

The following are the most common errors you might encounter when using the Oracle Database adapter, along with their probable cause and resolution.

- [Error in loading the adapter bindings](#)
- [The Oracle database adapter does not display in the list of adapters in BizTalk Server Administration console](#)
- [Error while retrieving XML output with more than 65,536 nodes](#)
- [Error while performing operations on the Oracle database](#)
- [XmlReaderParsingException due to an incorrect operation name in the specified action](#)
- [Error while specifying a connection URI for a WCF-Custom port in BizTalk](#)
- [Invalid cursor exception while invoking stored procedures that take REF CURSOR parameters](#)
- [Error while validating the response for the ReadLOB operation using BizTalk Server](#)
- [Schema validation may fail in polling scenarios](#)
- ['Unreasonable conversion requested' error when executing stored procedures with Record Types as parameters](#)
- [The adapter does not recognize the action on the physical port even though you use the binding file generated by the Consume Adapter Service add-in to create the ports](#)
- [Adapter throws an overflow exception \("System.OverflowException"\) on executing an operation](#)
- [Error with RootNode TypeName in BizTalk Projects](#)
- [Invalid binding warning when using the adapter in Visual Studio 2010](#)
- [BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host](#)
- [Memory usage and thread count increases when using the adapter in a transacted inbound operation](#)

Error in loading the adapter bindings

Problem

When you try to start the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in, you get the following error:

There was an error loading the binding, <binding name>, from your system configuration. ConfigurationErrorsException: Exception has been thrown by the target of an invocation.

Cause

When you try to start the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, WCF loads the adapter bindings for all the installed adapters. In turn, the adapter bindings are dependent on the specific client software for the enterprise application. You might face this issue for one or both of the following reasons:

- The required LOB client software is not installed on the computer where you installed the adapter.
- You did a Typical or Complete installation of the adapter, which installs all the adapters contained in the BizTalk Adapter Pack. However, the LOB client libraries might be installed for only one enterprise application. As a result, the GUI fails to load the bindings for the other adapters.

Resolution

- Make sure that the required LOB client versions are installed on the computer where you installed the BizTalk Adapter Pack. For information about the supported client versions, see [Frequently Asked Questions](#).
- Make sure you do a custom installation of the adapters to install only the adapter you need.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see the ODP.NET FAQ at <http://go.microsoft.com/fwlink/?LinkId=92834>.

The Oracle database adapter does not display in the list of adapters in BizTalk Server Administration console

Problem

Unlike the earlier version of the adapters shipped with BizTalk Server, the Oracle Database adapter shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console.

Cause

The latest Oracle Database adapter is a WCF custom binding. So, although the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF

custom bindings and hence, does not display the WCF-based Oracle Database adapter.

Resolution

You can explicitly add the Oracle Database adapter to the BizTalk Server Administration console by following the steps mentioned in [Adding the Oracle Database Adapter to BizTalk Server Administration Console](#).

Error while retrieving XML output with more than 65,536 nodes

Problem

The adapter gives the following error when retrieving XML output that has more than 65,536 nodes.

```
Maximum number of items that can be serialized or deserialized in an object graph is '65536'.Change the object graph or increase the MaxItemsInObjectGraph quota.
```

Cause

The adapter cannot serialize and deserialize an object with more than 65,536 items.

Resolution

You can fix this issue by setting the `maxItemsInObjectGraph` parameter. You can set this in either of the following two ways:

- Set this parameter by changing the `maxItemsInObjectGraph` parameter in the `ServiceBehavior` attribute on your service class.
- Add the following to your application's `app.config` file.

```
<behaviors>
  <endpointBehaviors>
    <behavior name="NewBehavior">
      <dataContractSerializer maxItemsInObjectGraph="65536000" />
    </behavior>
  </endpointBehaviors>
</behaviors>
```

A sample `app.config` looks like this.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.serviceModel>
    <behaviors>
      <endpointBehaviors>
        <behavior name="NewBehavior">
          <dataContractSerializer maxItemsInObjectGraph="65536000" />
        </behavior>
      </endpointBehaviors>
    </behaviors>
    <client>
      <endpoint behaviorConfiguration="NewBehavior" binding="oracleDBBinding"
        contract="IOutboundContract" name="oracle_ICalculator" />
    </client>
  </system.serviceModel>
</configuration>
```

Error while performing operations on the Oracle database

Problem

The adapter gives the following error when performing any operation on the Oracle database using BizTalk Server.

- **For BizTalk Server 2010**

```
System.ArgumentNullException: Value cannot be null.
```

Cause

The WCF action for the message is not specified. WCF requires a SOAP action to be specified for every operation, which informs the adapter about the operation to be performed on the LOB application.

Resolution

Specify the SOAP action in the send port or as a message context property in a BizTalk orchestration. For instructions, see [Specifying SOAP Action](#). See [Technical Reference](#) to see a list of actions for each operation.

XmlReaderParsingException due to an incorrect operation name in the specified action

Problem

The BizTalk Server Administration Console gives the following error when sending messages to an Oracle database:

```
Microsoft.ServiceModel.Channels.Common.XmlReaderParsingException: Invalid argument:
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="<operation_name>" Action="<action>" />
</BtsActionMapping>
```

Cause

If you configure a WCF-Custom port by importing the port binding file created by the Consume Adapter Service BizTalk Project Add-in, the action in the port is specified in the following format:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="http://MyService/Svc/Op1" />
</BtsActionMapping>
```

In the above format, the operation name is governed by the operation you chose while generating the schema. For example, if you generated schema for the Insert operation on a table, the operation name in the action will be "Insert". However, the operation name in the logical port created in the BizTalk orchestration in Visual Studio might be different.

Resolution

Make sure the operation names in both the logical port (in the BizTalk orchestration in Visual Studio) and the physical port (in BizTalk Server Administration Console) are same.

Error while specifying a connection URI for a WCF-Custom port in BizTalk

Problem

BizTalk Server gives the following error when you specify a connection URI to connect to the Oracle database.

```
Error saving properties.
(System.ArgumentException) The specified address is invalid.
(System.ArgumentException) Invalid address;
"<connection URI>" is not a well-formed absolute uri.
```

Cause

The connection URI does not adhere to the standard encoding format. For example, the value for a parameter might contain a space.

Resolution

Make sure the connection URI you specify adheres to the standard encoding format. For example, a blank space must be replaced by "%20".

Invalid cursor exception while invoking stored procedures that take REF CURSOR parameters

Problem

When you invoke procedures in the Oracle database that take REF CURSOR inputs, you might get the following exception:

```
Microsoft.ServiceModel.Channels.Common.TargetSystemException: ORA-01001: invalid cursor ---> Oracle.DataAccess.Client.OracleException
```

Cause

The PL/SQL block for the procedure that you are invoking could be piping the REF CURSORS, that is, the IN REF CURSOR could be getting assigned to the OUT REF CURSOR.

Resolution

The PL/SQL block must not pipe the IN to the OUT REF CURSORS without proper processing.

Error while validating the response for the ReadLOB operation using BizTalk Server

Problem

While performing a ReadLOB operation using the Oracle Database adapter with BizTalk Server, the response from the Oracle database fails validation against the Web Services Description Language (WSDL).

Cause

The WSDL contains a StreamBody node name that is defined for execution of service-based requests, but is not needed for BizTalk scenarios. Therefore, when the output XML, which does not contain the StreamBody node, is compared with the WSDL, validation fails.

Resolution

Remove the StreamBody node from the WSDL when validating against the output that was generated using BizTalk Server. Perform the following steps to do so:

1. The WSDL containing the StreamBody node looks like this.

```
<xs:element name="ReadLOBResponse">
  <xs:annotation>
    <xs:documentation>
      <doc:action xmlns:doc="http://schemas.microsoft.com/serviceModel/adapters/metadata/documentation">http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/TBL_ALL_DATATYPES/ReadLOB/response</doc:action>
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ReadLOBResult" nillable="true" type="ns3:StreamBody" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Replace the preceding with the following.

```
<xs:element name="ReadLOBResponse">
  <xs:annotation>
    <xs:documentation>
      <doc:action xmlns:doc="http://schemas.microsoft.com/serviceModel/adapters/metadata/documentation">http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/TBL_ALL_DATATYPES/ReadLOB/response</doc:action>
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="ReadLOBResult" type="xs:base64Binary" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

In this step, you removed the reference to type="ns3:StreamBody" in the original XSD and replaced it with type="xs:base64Binary". Also, you removed the nillable="true" value from the original XSD.

2. Remove the following from the WSDL.

```
<xs:complexType name="StreamBody">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Stream">
      <xs:simpleType>
        <xs:restriction base="xs:base64Binary">
          <xs:minLength value="0" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="StreamBody" nillable="true" type="ns3:StreamBody" />
```

Note

ReadLOB operation is not supported with BizTalk Server. You should use a table Select operation to read LOB data from a BizTalk Server solution.

Schema validation may fail in polling scenarios

Problem

Schema validation fails in scenarios where the Oracle Database adapter polls database tables that contain fields of type ROWID or UNROWID.

Cause

At design-time, when the adapter generates metadata for the table containing fields of type ROWID or UNROWID, the schema includes "nillable=false", which means that fields of type ROWID or UNROWID cannot be null. However, at run-time when the adapter retrieves the metadata, the fields of type ROWID or UNROWID contain null values. Hence the schema validation fails.

Resolution

If you are using the Oracle Database adapter with BizTalk Server, you may choose to disable schema validation. Alternatively, you may manually edit the schema to change "nillable=true" for ROWID and UNROWID data types.

'Unreasonable conversion requested' error when executing stored procedures with Record Types as parameters

Cause

Consider a scenario where an Oracle stored procedure takes a Record Type as a parameter. Assume that the Record Type is declared as <table name>%ROWTYPE, where the table has a column of LONG data type. When the Oracle Database adapter encounters the LONG data type, it sets the size of the data type equal to the value specified for the **LongDatatypeColumnSize** binding property. However, the Oracle database does not define a size for the LONG data type. So, when the adapter invokes the stored procedure, it results in an 'Unreasonable conversion requested' error.

Resolution

If a Record Type has a LONG data type, you must explicitly define it as part of a package.

The adapter does not recognize the action on the physical port even though you use the binding file generated by the Consume Adapter Service add-in to create the ports

Problem

After you use the Consume Adapter Service Add-in to generate schema for a specific operation on the Oracle database, the add-in also creates a port binding file. You can import this binding file using the BizTalk Server Administration console to create physical ports in BizTalk Server. However, when you send messages to the Oracle database using such ports, the adapter fails to understand the action specified on the port and gives an error similar to the following:

```
Microsoft.ServiceModel.Channels.Common.UnsupportedOperationException: Incorrect Action
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="<op_name>" Action="<action>" />
</BtsActionMapping>. Correct the specified Action, or refer to the documentation on the allowed formats for the Actions.
```

Cause

When you create logical ports in a BizTalk orchestration, you specify certain names for the operations on those ports or you just use the default names like Operation_1, Operation_2, etc. However, in the binding file generated by the Consume Adapter Service Add-in, the operation name is same as the name of the Oracle database operation for which you generate metadata. For example, if you generate metadata for Select operation on ACCOUNTACTIVITY table in the Oracle database, the action will be set to the following:

```
<Operation Name="Select" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY/Select" />
```

When you import the binding file, the same action is set on physical port. So, the operation names on the logical port (Operation_1, Operation_2, etc.) do not match the operation names specified in the action on the physical port, resulting in an error.

Resolution

Make sure the operation name in the logical port is the same as the operation name specified as part of the action in the physical port. Do one of the following:

- Change the operation name in the logical port in BizTalk orchestration from Operation_1, etc. to the operation for which you generate metadata, for example Select.
- Change the operation name in the action on the physical port to the operation name in the logical port. For example, you could change the action in the physical port to resemble the following:

```
<Operation Name="Operation_1" Action="http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/ACCOUNTACTIVITY/Select" />
```

Adapter throws an overflow exception ("System.OverflowException") on executing an operation

Problem

Using the adapter, if you try to perform an operation containing Oracle numeric data types inside DataSets or weakly-typed REF CURSORS, the adapter might throw an overflow exception.

Cause

This happens if you supply a large value for the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS that cannot fit into the respective .NET type.

Resolution

If you want to pass large values for the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS, you must enable safe typing by setting the value of the **EnableSafeTyping** binding property to **true**. Enabling safe typing exposes the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS as strings.

Error with RootNode TypeName in BizTalk Projects

Problem

In a BizTalk project in Visual Studio, if the schemas generated from the Consume Adapter Service Add-in contains invalid characters or reserved words for the **RootNode TypeName** property, the following error will occur while compiling the project:

```
Node <node reference> - Specify a valid .NET type name for this root node.
The current .NET type name of this root node is invalid (it is a reserved BizTalk keyword or is an invalid C# identifier).
```

Resolution

1. Right-click the root node referenced in the error and select **Properties**.
2. For the **RootNode TypeName** property, remove any illegal characters or reserved words, for example, dot (.).

Invalid binding warning when using the adapter in Visual Studio 2010

Problem

When you use the adapter to create an application in Visual Studio 2010 and you open the configuration file (app.config) generated by the adapter, you see a warning similar to the following:

```
The element 'bindings' has invalid child element 'oracleDBBinding'. List of possible elements expected: 'basicHttpBinding, customBinding, ...
```

Cause

This warning appears because the Oracle Database adapter binding, `oracleDBBinding`, is not a standard binding shipped with the Windows Communication Foundation (WCF).

Resolution

You can safely ignore this warning.

BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host

Problem

BizTalk Server throws an XLANG exception or an exception stating that the application cannot locate the document specification because multiple schemas matched the message type.

Cause

This happens because of either of the following:

- You have generated more than one Notification schema in a BizTalk Server project, deployed it to a BizTalk Server application, and then ran the application to receive notifications from the Oracle database. Because the Notification schemas are common, there is a conflict between the schemas that are deployed in the BizTalk Server application.
- In case of multiple projects, you have generated a Notification schema for each of the BizTalk Server projects, deployed each project to a separate BizTalk Server application on the same host, and then ran an application or applications to receive notifications from the Oracle database. Because the schemas and assemblies are accessible across the applications in BizTalk Server, there is a conflict between the common schemas deployed under various BizTalk Server applications and assemblies.

Resolution

Use a single Notification schema file for a BizTalk Server application. If you need to use the Notification schema in multiple BizTalk Server applications on the same host, create an application containing a single Notification schema, and then use the notification schema from all other applications in BizTalk Server.

Memory usage and thread count increases when using the adapter in a transacted inbound operation

Problem

In a transacted inbound operation, such as Polling, **if there is no data available in the table being polled** and the adapter continues to poll, over a period of time you experience an increase in the memory usage and the thread count.

Cause

If there is no data available in the table being polled, after every receive timeout cycle, Windows Communication Foundation (WCF) spawns a new thread to continue the polling operation. Hence, the thread count and memory usage increases over a period of time. However, if the table being polled has some data, the same thread continues to perform all subsequent polls.

Resolution

We recommend setting the **ReceiveTimeout** to the maximum possible value, which is 24.20:31:23.6470000 (24 days) so that a new thread is spawned only every 24 days. This will ensure that the memory usage and thread count does not grow too much too soon.

For more information about the **ReceiveTimeout** binding property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For instructions on specifying binding properties, see [Specifying Binding Properties](#).

Note

When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.

See Also

Tasks

[Troubleshooting Installation Issues](#)

Other Resources

[Troubleshooting](#)

Troubleshooting Performance Issues

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve performance issues that you might encounter when using Microsoft BizTalk Adapter for Oracle Database.

Known Issue

The following is the most common performance issue you might encounter when using the Oracle Database adapter, along with its probable cause and resolution.

- [Slowdown or stall in throughput when using the adapter with BizTalk Server](#)

Slowdown or stall in throughput when using the adapter with BizTalk Server

Problem

When using the Oracle Database adapter with BizTalk Server, the number of messages sent or received by the adapter slows down or comes to a stall.

Cause

The **EnableBizTalkCompatibilityMode** binding property is not set on the WCF-Custom send or receive port in BizTalk Server Administration console.

Resolution

Set the **EnableBizTalkCompatibilityMode** binding property to True. For more information about this property, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). For instructions on how to set a binding property, see [Specifying Binding Properties](#).

Possible memory leak on a 64-bit computer when using the Oracle database adapter to perform operations involving FLOAT data type

Problem

You may experience a memory leak when using the Oracle Database adapter on a 64-bit computer to perform operations that involve FLOAT data types.

Resolution

Install .NET <version> (x64) on the 64-bit computer.

See Also

Other Resources

[Troubleshooting](#)

Using Performance Counters

Microsoft BizTalk Adapter Pack clients can use performance counters to gauge the performance of the adapters. The BizTalk Adapter Pack setup program creates the performance counter category **BizTalk .NET Adapter for Oracle DB** along with installing the BizTalk Adapter Pack.

The LOB Time (Cumulative) Performance Counter

The **BizTalk .NET Adapter for Oracle DB** category has one performance counter called the "LOB Time (Cumulative)." This performance counter denotes the time, in milliseconds, that the LOB client library takes to complete an action that the adapter initiates. The Oracle Database adapter creates an instance of the performance counter in any of the following patterns:

```
<process id>:<app domain id>:<oracle data source>:<string>
```

Where "string" could be:

- Connection.Open
- Connection.Close
- Metadata
- Message action. For example, if the action is `http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert` then the string will be `SCOTT.Table.EMP.Insert`.

The Oracle data source is the same as specified in the connection URI.

The performance counter is initialized only after the adapter makes the first call to the Oracle database. Also, the InstanceLifetime property of the performance counter is set to 'Process', which means that the performance counter ceases to exist as soon as the program that creates the counter terminates. For more information about the InstanceLifetime property, see <http://go.microsoft.com/fwlink/?LinkId=104181>.

Note

The precision of the LOB Time (Cumulative) performance counter is 16 milliseconds.

Enabling Performance Counters

The performance counters can be enabled or disabled by setting the binding property **EnablePerformanceCounters**. To enable performance counters, set the **EnablePerformanceCounters** binding property to **True**. To disable performance counters, set **EnablePerformanceCounters** to **False**. By default, **EnablePerformanceCounters** is set to **False**.

Performance Counters and the WCF LOB Adapter SDK

Changing the value of the **EnablePerformanceCounters** binding property also changes the value of the corresponding performance counter for the WCF LOB Adapter SDK. Also, the binding property for the WCF LOB Adapter SDK is static, whereas that for the BizTalk Adapter Pack is dynamic. Therefore, if there are two instances of the Oracle Database adapter binding in the AppDomain, and the **EnablePerformanceCounters** binding property is set to **True** in one and **False** in the other, the adapter-specific performance counter will be enabled in one and disabled in the other. However, because the binding property for WCF LOB Adapter SDK is static, it will either be set to **True** or **False** depending on what value was specified last.

See Also

Other Resources

[Troubleshooting](#)

Exceptions and Error Handling

This section lists the exceptions that the Microsoft BizTalk Adapter for Oracle Database throws. These can contain:

- An inner exception, which is a system exception that the .NET Framework throws.
- An LOB exception that the LOB client library throws.

For more information about the inner exception, see the respective .NET Framework or Oracle documentation. Exceptions also contain a detailed error message that helps in resolving the problem.

Exception	Possible Cause/Description
XmlReaderParsingException	The adapter throws this exception if it does not support the specified type, or if an incorrect value is specified for the type. Also, the input XML might be incorrect. An incorrect value includes cases where the maximum amount of text or maximum digits is exceeded. The input XML might be incorrect if the operation name or namespace is incorrect.
UnsupportedOperationException	The adapter throws this exception when the adapter client specifies an invalid action.
ArgumentException	The adapter throws this exception if an incorrect value is specified for an argument.
NotImplementedException	The adapter throws this exception if some method in the XMLReader reader is not implemented.
ArgumentNullException	The adapter throws this exception if a required argument is not specified.
ArgumentOutOfRangeException	The adapter throws this exception if it tries to access a non-existent entity or out-of-range entity.
XmlReaderGenerationException	The adapter throws this exception when it is unable to generate an XmlReader from the output message.
MetadataException	The adapter throws this exception if there is an error during metadata retrieval, browse, or search.
CredentialException	The adapter throws this exception if there is a problem retrieving or using security tokens or if the required credentials are not specified.
InvalidUriException	The adapter throws this exception if the connection URI does not have the required components for the connection string.

ConnectionException	The adapter throws this exception if there is a problem connecting to the Oracle database using ODP.NET. The inner exception contains the Oracle exception.
TimeoutException	The adapter throws this exception if the timeout specified for an operation is lapsed. The inner exception contains the specifics of why the specified timeout was not sufficient.
ListenerException	The adapter throws this exception if there is a problem in receiving a message from the target system. This message denotes a problem related to the Oracle listener. The inner exception has the specifics of the issue.
TargetSystemException	The adapter throws this exception if Oracle returns an error or invalid response. The inner exception contains the Oracle runtime exception.
InvalidOperationException	The adapter throws this exception if adapter tries to perform an invalid operation on the target system. The inner exception contains the specifics of the invalid operation being performed.
OverflowException	The adapter throws this exception if while performing operation containing Oracle numeric data types inside DataSet or weakly-typed REF CURSORS, a large value is specified for these Oracle numeric data types that cannot fit into the respective .NET type.

See Also

Other Resources

[Troubleshooting](#)

Technical Reference

This section covers technical reference information about Microsoft BizTalk Adapter for Oracle Database.

In This Section

- [Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)
- [Metadata Node IDs](#)
- [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Oracle Database](#)
- [Glossary](#)

Messages and Message Schemas for BizTalk Adapter for Oracle Database

The Microsoft BizTalk Adapter for Oracle Database is a Windows Communication Foundation (WCF) custom binding. It exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel.

As a WCF service, the Oracle Database adapter exposes metadata for its operations and data types by using standard WCF mechanisms. The sections in this topic describe the XML structure of the messages and data types that the Oracle Database adapter uses.

In This Section

- [Basic Oracle Data Types](#)
- [Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views](#)
- [Message Schemas for Special LOB Operations](#)
- [Message Schemas for Functions and Procedures](#)
- [Message Schemas for REF CURSORS](#)
- [Message Schemas for RECORD Types](#)
- [Message Schemas for the SQLEXECUTE Operation](#)
- [Message Schemas for the Polling Operations](#)
- [Message Schemas for the Composite Operation](#)
- [Message Schemas for the Notification Operation](#)

See Also

Other Resources

[Technical Reference](#)

Basic Oracle Data Types

This topic describes how the Microsoft BizTalk Adapter for Oracle Database surfaces basic Oracle data types.

Supported Oracle Data Types

The Microsoft BizTalk Adapter for Oracle Database supports safe typing for some Oracle data types. When safe typing is enabled, these data types are represented as strings. You configure safe typing by setting the **EnableSafeTyping** binding property. Safe typing is disabled by default. For more information about the Oracle Database adapter binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#).

Note

Safe typing is not supported if data types are inside User Defined Types (UDTs).

The following table shows how the Oracle data types are surfaced with safe typing disabled (**EnableSafeTyping** is false). Oracle data types that are affected by the **EnableSafeTyping** binding property are marked with an asterisk (*).

Oracle Data Type	XSD type	.NET type	Comments
BFile	input: xsd:string output: xsd:base64Binary	String Byte[]	BFile data type is not supported inside complex types (such as RecordType, TableType, UDT, and VArray).
Blob	xsd:base64Binary	Byte[]	Supported for table operations and procedures.
Char	xsd:string	String	Supported for table operations and procedures.
Clob	xsd:string	String	Supported for table operations and procedures.
Date* (No safe typing if inside an UDT)	xsd:dateTime	DateTime	Date values cannot contain time zone information (UTC or UTC offsets): <ul style="list-style-type: none"> xsd:dateTime values must not contain UTC or UTC offsets DateTime.Kind must be DateTimeKind.Unspecified <p>If time zone information is specified, the adapter throws an XmlReaderParsingException exception with a message that indicates the field.</p>
Float**	xsd:float if prec <= 7 xsd:double if prec > 7 and <= 15 xsd:string if prec > 15	Float Double String	-

IntervalYM	xsd:string xsd:long if inside an UDT	String Long if inside an UDT	The value should be expressed in Oracle native format: Year-Month; For example, "1-2" (1 year and 2 months).
IntervalDS	xsd:string xsd:duration if inside an UDT	String TimeSpan if inside an UDT	The value should be expressed in Oracle native format: Day HH:MI:SSxFF; for example, "5 15:30:12.99"
Long	xsd:string	String	Supported for all table operations, stored procedures, and functions. 📌Note Starting with the Oracle database 9i release, the LONG data type is deprecated. Oracle recommends using the LOB data types instead. Hence, when performing operations on the Oracle database using the Oracle Database adapter, we recommend using Oracle database artifacts that operate on LOB data types and not the LONG data type.
LongRaw	xsd:base64Binary	Byte[]	-
NChar	xsd:string	String	-
NClob	xsd:string	String	Supported for table operations and procedures.
Number**	xsd:decimal if prec <= 28 xsd:string if prec > 28	Decimal String	-
NVarchar2	xsd:string	String	-
Raw	xsd:base64Binary	Byte[]	Supported for table operations and procedures.
RowID	xsd:string	String	-
TimeStamp* (No safe typing if inside an UDT)	xsd:dateTime if prec <= 7 xsd:string if prec > 7	DateTime String	TimeStamp values cannot contain time zone information (UTC or UTC offsets): <ul style="list-style-type: none"> • xsd:dateTime values must not contain UTC or UTC offsets • DateTime.Kind must be DateTimeKind.Unspecified <p>If time zone information is specified, the adapter throws an XmlReaderParsingException exception with a message that indicates the field.</p>

TimeStampLTZ	xsd:string	String	TimeStampLTZ is not supported inside UDTs. Outside an UDT: The value should be expressed in NLS_TIMESTAMP_TZ_FORMAT.
TimeStampTZ	xsd:string xsd:date Time if inside an UDT	String DateTime if inside an UDT	Outside an UDT: The value should be expressed in NLS_TIMESTAMP_TZ_FORMAT.
Decimal*	xsd:decimal if prec <=28 xsd:string if prec > 28	Decimal String	-
Varchar2	xsd:string	String	-
Binary Float**	xsd:float if prec <=7 xsd:string if prec > 7	Float String	You must specify the value in a form consistent with your locale (System.Globalization.CultureInfo.CurrentCulture). For example, for English locale use a period character ('.') to specify the decimal; for French locale, use a comma character (',').
Binary Double**	xsd:double if prec <=15 xsd:string if prec > 15	Double String	-
Binary Integer**	xsd:integer	Int32	Supported for procedures, functions, and packages.
Boolean	xsd:boolean	Nullable boolean	
XMLTYPE	xsd:string	String	Supported for top level procedure parameters. Reserved XML characters like '<', '>' must be replaced with their entity representation (&lt; , &gt;) when developing applications in BizTalk, and when using WCF channel Model. This is not required in the case of WCF Service Model.

*The way in which these Oracle data types are surfaced is affected by the **EnableSafeTyping** binding property.

The way in which these Oracle numeric data types inside DataSets and weakly-typed REF CURSORS are surfaced is affected by the **EnableSafeTyping binding property.

◆ Important

- The maximum length of the value in an Oracle data type in the Oracle Database adapter is bound by the maximum length of the value supported by ODP.NET for the Oracle data type.
- The Oracle Database adapter internally treats the Oracle numeric data types inside UDTs as .NET Decimal. However, in general (that is outside UDTs), the Oracle Database adapter internally treats the Oracle numeric data types as OracleDecimal.

Safe Typing Enabled

The following table shows how the Oracle data types that are affected by safe typing are changed when the **EnableSafeTyping** binding property is true.

Oracle Data Type	XSD type	.NET type	Comment
Date	xsd:string	String	The value should be expressed in Oracle NLS_DATE_FORMAT.
TimeStamp	xsd:string	String	The value should be expressed in Oracle NLS_TIMESTAMP_FORMAT.

◆ Important

If safe typing is enabled, the Oracle numeric data types inside DataSets and weakly-typed REF CURSORS are always exposed as strings.

Oracle data types that are not in this table are surfaced in the same way whether safe typing is enabled or disabled.

Validation

The Microsoft BizTalk Adapter for Oracle Database performs no explicit validation on the values that you specify for Oracle data types. However, depending on the Oracle data type and whether safe typing is enabled or disabled, implicit validation may be performed:

- When de-serializing between the XML passed in a message and the .NET types that are used internally by the adapter.
- By ODP.NET for some data types.

See Also

Other Resources

[Technical Reference](#)

Message Schemas for the Basic Insert, Update, Delete, and Select Operations on Tables and Views

The Microsoft BizTalk Adapter for Oracle Database surfaces basic Insert, Update, Delete, and Select operations for each table and view in the Oracle database. These operations perform the appropriate SQL statement qualified by a WHERE clause. The Oracle Database adapter uses strongly-typed records and record sets in these operations.

Message Structure for Basic Table Operations

The following table shows the XML message structure for the basic table operations exposed by the Oracle Database adapter on Oracle database tables. The target table for an operation is specified in the message action and also appears in the target namespace.

Operation	XML Message	Description	SQL Executed by the Adapter
Insert	<p>Multiple Record Insert:</p> <pre> <Insert xmlns="[VERSION] /[SCHEMA]/Table/[TABLE_NAME]"> <RECORDSET > <[TABLE_NAME]RECORDINSERT> <[FIELD1_NAME InlineValue="value1">value1</[FIELD1_NAME]> <[FIELD2_NAME InlineValue="value2">value2</[FIELD2_NAME]> ... </[TABLE_NAME]RECORDINSERT> <[TABLE_NAME]RECORDINSERT > <[FIELD1_NAME InlineValue="value1">value1</[FIELD1_NAME]> <[FIELD2_NAME InlineValue="value2">value2</[FIELD2_NAME]> ... </[TABLE_NAME]RECORD </pre>	<p>There are two types of Insert operations:</p> <ul style="list-style-type: none"> Multiple Record Insert inserts the supplied record set of strongly-typed data into the target table. Bulk Insert inserts the record set returned by a SELECT query specified in the QUERY element into the target table. This is done by using the comma-separated list of columns specified in the COLUMN_NAMES element. <p>A message can contain only one kind of Insert operation.</p>	<ul style="list-style-type: none"> Multiple Record Insert <div data-bbox="1177 748 1497 972" style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>INSERT INTO TABLE_NAME (FIELD1_NAME, FIELD2_NAME, ...)VALUES (value1, value2, ...);</pre> </div> Bulk Insert <div data-bbox="1177 1066 1497 1256" style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>INSERT INTO TABLE_NAME (COLUMN_list) SELECT_query;</pre> </div>

	<pre>INSERT> ... </RECORDSE T> </Insert></pre>		
	<p>Bulk Insert:</p> <pre><Insert xmlns=" [VERSION] /[SCHEMA]/Ta ble/[TABLE_N AME]"> <COLUMN_NA MES>[COLUMN_ list]</COLUM N_NAMES> <QUERY>[SE LECT_query]< /QUERY> </Insert></pre>		
Insert Response	<pre><InsertResponse xmlns=" [VERSION]/[SC HEMA]/Table/ [TABLE_NAME] "> <InsertResult>[rows in serted]</Ins ertResult> </InsertResp onse></pre>	The number of rows inserted is returned in the InsertResult element.	--
Select	<pre><Select xmlns=" [VERSION] /[SCHEMA]/Ta ble/[TABLE_N AME]"> <COLUMN_NA MES>[COLUMN_ list]</COLUM N_NAMES> <FILTER>WH ERE_clause</ FILTER> </Select></pre>	A SELECT query is performed on the target table using the WHERE clause specified in the FILTER element. The result set contains the columns in the comma-separated list of column names specified in the COLUMN_NAMES element.	<pre>SELECT COLUMN_list F ROM TABLE_NAME WHERE WHERE_clause;</pre>

Select Response	<pre> <SelectResponse xmlns=" [VERSION]/[SCHEMA]/Table /[TABLE_NAME]"> <SelectResult> <[TABLE_NAME]RECORDS ELECT> <[FIELD1_NAME]>value1</[FIELD1 _NAME]> <[FIELD2_NAME]>value2</[FIELD2 _NAME]> ... </[TABLE_NAME]RECORD SELECT> </SelectResult> </SelectResponse> </pre>	The result set generated by the SELECT query.	--
Update	<pre> <Update xmlns="[VERSION] /[SCHEMA]/Table/[TABLE_N AME]"> <RECORDSET > <[FIELD1 _NAME]>value 1</[FIELD1_N AME]> <[FIELD2 _NAME]>value 2</[FIELD2_N AME]> ... </RECORDSE T> <FILTER>WH ERE_clause</ FILTER> </Update> </pre>	Rows that match the where clause specified in the FILTER element are updated to the values specified in the RECORDSET. Only the columns that are specified in the RECORDSET are updated in each matching row.	<pre> UPDATE [TABLE_NAME] SET [FIELD1_NAME] = value1, [FIELD2_NAME] = value2, ... WHERE WHERE_clause; </pre>

Update Response	<pre><UpdateResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <UpdateResult>[rows inserted]</UpdateResult> </UpdateResponse></pre>	The number of rows updated is returned in the UpdateResult element.	--
Delete	<pre><Delete xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <FILTER>WHERE_clause</FILTER> </Delete></pre>	Rows matching the WHERE clause specified by the FILTER element are deleted.	DELETE FROM [TABLE_NAME] WHERE WHERE_clause;
Delete Response	<pre><DeleteResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <DeleteResult>[rows inserted]</DeleteResult> </DeleteResponse></pre>	The number of rows deleted is returned in the DeleteResult element.	--

[VERSION] = The message version string; for example, http://Microsoft.LobServices.OracleDB/2007/03.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = Name of the table; for example, EMP.

[FIELD1_NAME] = Table field name; for example, EMPNAME.

[COLUMN_list] = Comma-separated list of columns; for example, NAME.

[SELECT_query] = A SQL SELECT statement specified in the QUERY element of a Bulk Insert operation; for example, "SELECT * from MyTable"

[WHERE_clause] = WHERE_clause for the SELECT statement used for the operation; for example, ID > 10.

◆ Important
The message structure for the basic table operations on views is the same as that on tables, but the namespace for the operation specifies a view rather than a table: <Insert xmlns = "[VERSION]/[SCHEMA]/View/[VIEW_NAME]">.

Message Actions for Basic Table Operations

The following table shows the message actions that are used by the Oracle Database adapter for the basic table operations on tables. The Oracle Database adapter uses the table name specified in the message action to determine the target table of the operation.

Operation	Message Action	Example
-----------	----------------	---------

Insert	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Insert	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert
Insert Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Insert/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert/response
Select	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Select	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select
Select Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Select/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select/response
Update	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Update	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Update
Update Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Update/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Update/response
Delete	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Delete	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete
Delete Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/Delete/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete/response

[VERSION] = The message version string; for example, http://Microsoft.LobServices.OracleDB/2007/03.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = Name of the table; for example, EMP.

◆ Important

The message action for an operation on a view is the same as that for a table except that "View" replaces "Table"; for example, http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/**View**/EMPVIEW/Insert.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for Special LOB Operations

The ReadLOB and UpdateLOB operations are surfaced for tables and views that contain LOB columns; that is columns that are used to store Oracle large object (LOB) data. These operations enable you to read or write the LOB data as a stream of base64Binary-encoded data. They operate on a single column of LOB data in a single row.

For an overview of the ReadLOB and UpdateLOB operations and of the Oracle LOB data types supported, see [Operations on Tables and Views That Contain LOB Data](#).

Message Structure of LOB Data-Type Operations

The following table shows the structure of the request and response messages for the ReadLOB and UpdateLOB operations. The target table for the operation is specified in the message action and also appears in the target namespace.

Operation	XML Message	Description
ReadLOB	<pre><ReadLOB xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <LOB_COLUMN>[COL_NAME]</LOB_COLUMN> <FILTER>[WHERE_clause]</LOB_COLUMN> </ReadLOB></pre>	<p>The LOB data in the</p> <ul style="list-style-type: none"> column identified by the LOB_COLUMN element, and the row that matches the where clause specified in the FILTER element <p>is returned.</p> <p>The where clause should match only a single row. If there is more than one matching row, the LOB data in the first matching row is returned.</p> <p>Important The ReadLOB operation is designed to support input streaming of LOB data in the WCF service model. You should use a table Select operation to read LOB data from a WCF Channel Model or BizTalk Server solution.</p>
ReadLOB Response	<pre><ReadLOBResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <ReadLOBResult> [LOB_DATA] </ReadLOBResult> </ReadLOBResponse></pre>	<p>The LOB data is returned as a stream of base64Binary encoded data.</p> <p>Important The WSDL returned by the adapter does not match the actual schema used by the adapter for the ReadLOB response message.</p>
UpdateLOB	<pre><UpdateLOB xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <LOB_COLUMN>[COL_NAME]</LOB_COLUMN> <FILTER>[WHERE_clause]</LOB_COLUMN> <Stream>[LOB_DATA]</Stream> </UpdateLOB></pre>	<p>The LOB data in the</p> <ul style="list-style-type: none"> column identified by the LOB_COLUMN element, and the row that matches the where clause specified in the FILTER element <p>is updated with the base64Binary encoded data in the stream.</p> <p>The where clause should match only a single row. If there is more than one matching row, an exception is thrown.</p> <p>Note The UpdateLOB operation replaces all of the data in the specified column and row.</p>

Update LOB Response	<UpdateLOBResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> </UpdateLOBResponse>	An empty response is returned.
---------------------	---	--------------------------------

[VERSION] = The message version string; for example, "http://Microsoft.LobServices/OracleDB/2007/03".

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = The table that contains the targeted LOB column; for example, EMP.

[COL_NAME] = The name of the targeted LOB column; for example, LOB_FIELD.

[WHERE_clause] = An Oracle database SELECT statement WHERE clause that matches a single row; for example, ID = 1.

[LOB_DATA] = The LOB column data in base64Binary type.

◆ Important

The message structure for the ReadLOB and UpdateLOB operations on views is the same as that on tables except that the namespace for the operation specifies a view rather than a table: <ReadLOB xmlns="[VERSION]/[SCHEMA]/View/[VIEW_NAME]">.

Message Actions for LOB Data-Type Operations

The following table shows the message actions that are used by the Oracle Database adapter for the ReadLOB and UpdateLOB operations on tables. The Oracle Database adapter uses the table name specified in the message action to determine the target table for the operation.

Operation	Action	Example
ReadLOB	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/ReadLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/ReadLOB
ReadLOB Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/ReadLOB/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/ReadLOB/response
UpdateLOB	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/UpdateLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/UpdateLOB
UpdateLOB Response	[VERSION]/[SCHEMA]/Table/[TABLE_NAME]/UpdateLOB/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/CUSTOMER/UpdateLOB/response

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.OracleDB/2007/03".

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = The table that contains the targeted LOB column; for example, CUSTOMER. (The SCOTT.CUSTOMER table is installed by a SQL script included in the samples.)

◆ Important

The message action for ReadLOB and UpdateLOB operations on views is similar to that used for tables, except that action for the operation specifies a view rather than a table: [VERSION]/[SCHEMA]/View/[VIEW_NAME]/ReadLOB.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for Functions and Procedures

The Microsoft BizTalk Adapter for Oracle Database surfaces Oracle database functions and stored procedures as operations. This section describes the message structure and actions used to invoke functions and procedures.

Message Structure of Functions and Procedures

The operations surfaced for functions and stored procedures follow a request-response message exchange pattern. The following table shows the structure of these request and response messages.

Operation	XML Message	Description
Stored Procedure Request	<pre><[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]></pre>	Supports Oracle IN and IN OUT parameters in the message body
Stored Procedure Response	<pre><[SP_NAME]Response xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure"> <[PRM1_NAME]>value1<[/[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]Response></pre>	Supports Oracle OUT and IN OUT parameters in the message body
Function Request	<pre><[FN_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Function"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[FN_NAME]></pre>	Supports Oracle IN and IN OUT parameters in the message body
Function Response	<pre><[FN_NAME]Response xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Function"> <[FN_NAME]Result>return_value</[FN_NAME]Result> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[FN_NAME]Response></pre>	Supports Oracle OUT and IN OUT parameters in the message body <ul style="list-style-type: none"> The function return value is returned in the <[FN_NAME]Result> element. This is the first element in the response message. It comes before any parameters.

Packaged Procedure or Function Request	<pre><[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]></pre>	Same as Function or Stored Procedure
Packaged Procedure or Function Response	<pre><[SP_NAME]Response xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]Response></pre>	Same as Function or Stored Procedure

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[SP_NAME] = The stored procedure to be executed; for example, SP_INSERT.

[FN_NAME] = The function to be executed; for example, FN_GETID.

[PRM1_NAME] = The name of the Oracle parameter. See the Description column for supported parameter directions for each message.

[PACKAGE_NAME] = The name of the package that contains the targeted procedure or function.

The Oracle database supports overloading for stored procedures and functions. The Oracle Database adapter supports this capability by appending an overload string to the target namespace for each overloaded artifact. The value of this string is "overload1" for the first overload, "overload2" for the second overload, and so on. The following example shows the message structure for two overloaded stored procedures.

<pre>Stored Procedure Overload 1: <[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/overload1"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value1</[PRM2_NAME]> ... </[SP_NAME]> Stored Procedure Overload 2: <[SP_NAME] xmlns="http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/overload2"> <[PRM1_NAME]>value1</I_[PRM1_NAME]> <[PRM2_NAME]>value1</I_[PRM2_NAME]> ... </[SP_NAME]></pre>

Message Actions of Functions and Procedures

The Oracle Database adapter uses the following message actions for stored procedure and function operations.

Message	Action	Example
Stored Procedure Request	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure/[SP_NAME]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure/SP_INSERT

Stored Procedure Response	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure/[SP_NAME]/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure/SP_INSERT/response
Function Request	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Function/[FN_NAME]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Function/FN_GETID
Function Response	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Function/[FN_NAME]/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Function/FN_GETID/response
Packaged Stored Procedure Request	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/CUSTOMER/SP_INSERT
Packaged Stored Procedure Response	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/CUSTOMER/SP_INSERT/response
Packaged Function Request	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[FN_NAME]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/CUSTOMER/FN_GETID
Packaged Function Response	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Package/[PACKAGE_NAME]/[FN_NAME]/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/CUSTOMER/FN_GETID/response
Overloaded Stored Procedure Request	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure/[SP_NAME]/[OVERLOAD]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure/SP_INSERT/overload1
Overloaded Stored Procedure Response	http://Microsoft.LobServices.OracleDB/2007/03/[SCHEMA]/Procedure/[SP_NAME]/[OVERLOAD]/response	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure/SP_INSERT/overload1/response

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[SP_NAME] = The stored procedure to be executed; for example, SP_INSERT.

[FN_NAME] = The function to be executed; for example, FN_GETID.

[PACKAGE_NAME] = The name of the package that contains the targeted procedure or function.

[OVERLOAD] = The Overload parameter. The possible values are overload1, overload2, and so on.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for REF CURSORS

A REF CURSOR is an Oracle PL/SQL data type that represents a pointer to a result set in the Oracle database. REF CURSOR types enable input and output streaming of data and are ideal for transferring large amounts of data to and from a PL/SQL code block. Microsoft BizTalk Adapter for Oracle Database provides support for passing strongly-typed and weakly-typed REF CURSOR parameters to PL/SQL procedures and functions as IN, OUT and IN OUT parameters.

In the Oracle database, a REF CURSOR type can be either strongly-typed or weakly-typed:

- A strongly-typed REF CURSOR is declared with a RETURN clause as in `TYPE StrongCurType IS REF CURSOR RETURN emp%ROWTYPE;`. A strongly-typed REF CURSOR variable can only represent a result set that contains data that matches the type with which its REF CURSOR type is declared. The Oracle Database adapter returns a strongly-typed result set for a strongly-typed REF CURSOR.
- A weakly-typed REF CURSOR is declared without a RETURN clause as in `TYPE WeakCurType IS REF CURSOR;`. Oracle also provides a special REF CURSOR type called `SYS_REFCURSOR` that can be used to declare weakly-typed REF CURSOR variables. Weakly-typed REF CURSOR variables can represent a result set that contains any kind of row data. The Oracle Database adapter returns a weakly-typed result set of generic records for a weakly-typed REF CURSOR.

IN REF CURSOR Parameters

There is no ODP.NET API to create a REF CURSOR on the Oracle server, so the Oracle Database adapter cannot provide the capability for a client program to create and maintain REF CURSOR variables.

The Oracle Database adapter does, however, enable a client to pass IN REF CURSOR parameters to functions or stored procedures by specifying a block of PL/SQL code that returns a REF CURSOR. The adapter uses this code to create and OPEN a REF CURSOR variable on the Oracle server; it then calls the function or stored procedure using this variable as the IN parameter.

The Oracle Database adapter represents IN REF CURSOR parameters as strings that contain the PL/SQL code block. Within this block, the location of the REF CURSOR is specified with a question mark (?). The PL/SQL code block can contain an OPEN-FOR statement that contains a SQL query; or it can contain a function or procedure call in which an opened REF CURSOR is returned in an OUT parameter.

The following shows how to specify an IN REF CURSOR by calling a stored procedure or function to open the REF CURSOR.

```
<[IN_REF_CURSOR_PARAM_NAME]>begin [SP_NAME]([SP_PARAMS...], ?, [SP_PARAMS...]); end;</[IN_REF_CURSOR_PARAM_NAME]>
```

Example:

```
<EMP_RC>begin GETEMP(1, ?); end; </EMP_RC>
```

The following shows how to specify an IN REF CURSOR by using a SELECT query to open the REF CURSOR.

```
<IN_REF_CURSOR_PARAM_NAME>begin open ? for select [FIELD_NAMES] from [TABLE_NAME]; end;</IN_REF_CURSOR_PARAM_NAME>
```

Example:

```
<EMP_RC>begin open ? for select * from EMP; end;</EMP_RC>
```

OUT REF CURSOR Parameters

OUT REF CURSOR parameters are returned as either strongly-typed or weakly-typed result sets. The type of the result set returned depends on whether the REF CURSOR parameter is declared as a strongly-typed or weakly-typed REF CURSOR in the stored procedure or function definition on the Oracle server. A strongly-typed result set is returned for strongly-typed REF CURSOR parameters and a weakly-typed result set is returned for weakly-typed REF CURSOR parameters (for example, parameters declared with a `SYS_REFCURSOR` type).

The following shows the XML for a strongly-typed OUT REF CURSOR parameter.

```
<[PARAM_NAME]>
```

```

<[PARAM_NAME]RECORD>
  <[COL1_NAME]>value1</[COL1_NAME]>
  <[COL2_NAME]>value2</[COL2_NAME]>
  ...
</[PARAM_NAME]RECORD>
</[PARAM_NAME]>

```

[PARAM_NAME] = OUT REF CURSOR parameter name; for example, EMP_REFCURSOR
[COL_NAME] = Name of a column in the REF CURSOR type; for example, Name.

The following shows the XML for a weakly-typed OUT REF CURSOR parameter.

```

<[PARAM_NAME]>
  <GenRecordRow xmlns="oracledb">
    <GenRecordColumn>
      <ColumnName>COL_NAME</ColumnName>
      <ColumnValue>COL_VALUE</ColumnValue>
      <ColumnType>COL_TYPE</ColumnType>
    </GenRecordColumn>
    ...
  </GenRecordRow>
  ...
</[PARAM_NAME]>

```

[COL_NAME] = Name of column; for example, Name
[COL_VALUE] = Column value; for example, Scott
[COL_TYPE] = Column data type; for example, System.String

IN OUT REF CURSOR Parameters

Because the Oracle Database adapter models IN REF CURSOR parameters as strings and OUT REF CURSOR parameters as complex types, it cannot support a single type for an IN OUT REF CURSOR parameter. For this reason, it treats IN OUT REF CURSOR parameters as two different parameters: an IN parameter in the request message and an OUT parameter in the response message.

To avoid a name conflict in service model programming, the Oracle Database adapter appends the string "_IN" to the IN parameter in the request message so that for a given parameter named [PARAM_NAME], two parameters are created:

- [PARAM_NAME]_IN is an IN REF CURSOR parameter in the stored procedure or function request message. It contains a PL/SQL statement (either a select query or a stored procedure or function call) that returns a REF CURSOR.
- [PARAM_NAME] is an OUT REF CURSOR parameter in the stored procedure or function response message. It contains the OUT REF CURSOR as a strongly-typed or weakly-typed result set.

◆ Important

The adapter cannot support a procedure or function that contains an IN parameter named [PARAM_NAME]_IN and an IN OUT REF CURSOR parameter named [PARAM_NAME]. This is because the adapter expects a parameter named [PARAM_NAME]_IN to represent the input to the REF CURSOR parameter, and you cannot specify both parameters in the request message.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for RECORD Types

Oracle RECORD types are structured PL/SQL data types that consist of one or more simple or structured database types. RECORD types are primarily used in PL/SQL stored procedures and functions to send and receive hierarchical data.

The Microsoft BizTalk Adapter for Oracle Database supports RECORD types in the following manner:

- RECORD types are surfaced as complex types.
- RECORD types can be nested (record in a record).
- RECORD types can be declared as TABLE%ROWTYPE parameters in stored procedures and functions.
- RECORD types can be declared as TYPE of RECORD parameters in PL/SQL packages; for example, `TYPE rec_type1 IS RECORD(name varchar2(100), age number(3));`.

Note

The Oracle Database adapter does not support BFILE types as RECORD members.

When a RECORD type parameter is used in a stored procedure or a function, it is qualified with the namespace of that operation. The following XML shows the structure of a RECORD type in a message:

```
<[REC_PARAM_NAME]>
  <[FIELD_NAME1] xmlns="[OPERATION_NAMESPACE]">value1</[FIELD_NAME1]>
  <[FIELD_NAME2] xmlns="[OPERATION_NAMESPACE]">value2</[FIELD_NAME2]>
  ...
</[REC_PARAM_NAME]>
```

[REC_PARAM_NAME] is the name of the RECORD parameter.

[FIELD_NAME] is the name of a field in the RECORD type.

[OPERATION_NAMESPACE] is the namespace of the stored procedure or function in which the RECORD parameter is being used.

The following XML shows the structure of a RECORD type parameter with a nested RECORD type field:

```
<[REC_PARAM_NAME]>
  <[FIELD_NAME1] xmlns="[OPERATION_NAMESPACE]">value1</[FIELD_NAME1]>
  <[FIELD_NAME2] xmlns="[OPERATION_NAMESPACE]">value2</[FIELD_NAME2]>
  <[REC_PARAM_NAME2]>
    <[FIELD_NAME1] xmlns="[OPERATION_NAMESPACE]">value1</[FIELD_NAME1]>
    <[FIELD_NAME2] xmlns="[OPERATION_NAMESPACE]">value1</[FIELD_NAME2]>
    ...
  </[REC_PARAM_NAME2]>
  ...
</[REC_PARAM_NAME]>
```

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for the SQLEXECUTE Operation

Microsoft BizTalk Adapter for Oracle Database surfaces strongly-typed metadata for artifacts present in the LOB system and exposes standard operations on these artifacts. However, there are scenarios where an application might require the execution of an arbitrary SQL statement that is driven by the business logic in the application. For example, you may want to:

- Perform an operation on database artifacts that are not surfaced by the Oracle Database adapter; for example, get the CURVAL or NEXTVAL of an Oracle SEQUENCE.
- Perform Data Definition Language operations; for example, create a table.
- Perform operations on a database artifact that was not present at design-time; for example, update records in a temporary table that is created by your business logic.
- Perform more complex DML operations on tables than the operations surfaced by the Oracle Database adapter; for example perform a query that includes a JOIN clause.

The Oracle Database adapter surfaces a special operation called the SQLEXECUTE operation to support such scenarios. By using this operation, you can specify an arbitrary SQL statement for the Oracle Database adapter to execute on the Oracle database. You can also specify multiple blocks of input parameters to the SQL statement. The Oracle Database adapter executes the SQL statement once for each set of parameters and returns any output as a generic (weakly-typed) record set.

Note

You can pass IN and IN OUT parameters to procedures, functions, and packages in the SQLEXECUTE operation. The invoked artifact will execute with the supplied parameters on the Oracle database; however, the SQLEXECUTE operation does not return the value of OUT and IN OUT parameters to the client. If you want to invoke procedures, functions, or packages, Microsoft recommends that you do so by invoking the dedicated operations that the Oracle Database adapter exposes for these Oracle artifacts.

The following XML shows the structure of the SQLEXECUTE operation:

```
<SQLEXECUTE xmlns="SQLEXECUTE">
  <SQLSTATEMENT> [STATEMENT] </SQLSTATEMENT>
  <PARAMETERSCHEMA>[PARAM_SPEC]</PARAMETERSCHEMA>
  <PARAMETERSET>
    <PARAMETERDATA>
      <PARAMETER xmlns:c="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <c:string>[PARAM_VAL_1]</c:string>
      </PARAMETER>
    </PARAMETERDATA>
    ...
  </PARAMETERSET>
</SQLEXECUTE>
```

[STATEMENT] = The SQL statement to execute; for example, "SELECT * from emp WHERE empno=:emp_no".

[PARAM_SPEC] = The list of the IN parameters in the SQL statement and their data types; for example, "emp_no NUMBER".

[PARAM_VAL_1] = The value of the first parameter.

Each <PARAMETERDATA> section contains a complete set of <PARAMETER> elements that match the schema in the <PARAMETERSCHEMA> section. The <PARAMETERSET> can contain multiple <PARAMETERDATA> sections. If this is the case, the SQL statement is executed multiple times, once against each parameter set.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for the Polling Operations

The Microsoft BizTalk Adapter for Oracle Database surfaces various inbound operations related to polling depending on the target object on the Oracle database. To poll tables and views, a single POLLINGSTMT operation is surfaced whereas each stored procedure, functions, and packaged procedures and functions are exposed as inbound operations for polling.

You can specify a **PollingId** parameter in the query string of the connection URI to qualify the namespace of the POLLINGSTMT operation. Setting this parameter only qualifies the namespace of the POLLINGSTMT operation; it does not change the message action. For more information about the Oracle Database adapter connection URI, see [The Oracle Database Connection URI](#).

You configure the polling operations by setting binding properties in the Oracle Database adapter. For more information about these binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). You set the **PollingStatement** binding property to specify a SQL statement, stored procedure, function or a procedure within a package for the polling query. The result set of this query is returned as data to your code in the polling operation.

Message Structure for the Polling Operations

The following table shows the XML message structure for the various polling operations.

Operation	Target Object	XML Message	Description
POLLINGSTMT	<ul style="list-style-type: none"> • Tables • Views 	<pre><?xml version="1.0" encoding="utf-8" ?> <POLLINGSTMT xmlns="[VERSION]/POLLINGSTMT[POLLING_ID]"> <POLLINGSTMTRECORD> <POLLINGSTMTRECORD> <FIELD1_NAME>val1</FIELD1_NAME> <FIELD2_NAME>val2</FIELD2_NAME> ... </POLLINGSTMTRECORD> ... </POLLINGSTMTRECORD> </POLLINGSTMT></pre>	<p>The structure of the result set contained in the POLLINGSTMTRECORD types is determined by the metadata that the adapter surfaces for the SQL SELECT query.</p> <p>The namespace of the POLLINGSTMT operation is determined by the PollingId parameter in the connection URI.</p>

<p>[CustomPollingOperation]</p>	<ul style="list-style-type: none"> • Stored Procedures • Functions • Packages 	<p>Stored Procedures</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/[SCHEMA]/PollingProcedure"> <[CustomPollingOperation] Result> <PRM1>[Value]</PRM1> <PRM2>[Value]</PRM2> ... </[CustomPollingOperation] Result> </[CustomPollingOperation]></pre> <p>Functions</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/[Schema]/PollingFunction"> <[CustomPollingOperation] Result> <COL1>[Value]</COL1> <COL2>[Value]</COL2> ... </[CustomPollingOperation] Result> </[CustomPollingOperation]></pre> <p>Packages</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/[Schema]/PollingPackage/[PACKAGE_NAME]/"> <[CustomPollingOperation] Result>[Value]</[CustomPollingOperation] Result> </[CustomPollingOperation]></pre>	<p>The structure of the result set in the polling operation is determined by the data type of the elements in the target object.</p>
---------------------------------	--	--	--

[Version] = <http://Microsoft.LobServices.OracleDB/2007/03>.

[CustomPollingOperation] = It is the same as the stored procedure, function, or packaged procedure or function name that are exposed as the inbound polling operation.

[Schema] = Name of the Oracle schema. For example, SCOTT.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for the Composite Operation

The Microsoft BizTalk Adapter for Oracle Database enables you to execute composite operations on the Oracle database. A composite operation can contain multiple operations, and in any order. For information about which operations can be included in a composite operation, see [Performing Composite Operations](#).

For information about how to perform composite operations using the Oracle Database adapter, see [Performing Composite Operations on Oracle Database by Using BizTalk Server](#).

Message Structure for the Composite Operation

Because a composite operation contains multiple individual operations; the message structure of a composite operation contains message structures of the individual operations. The composite operation message follows a request-response message exchange pattern.

The following table shows the structure of the request and response messages of a composite operation that contains an Insert operation, a packaged stored procedure that does not take any input parameters, and a Delete operation.

Operation	XML Message
Composite Operation Request	<pre> <?xml version="1.0" encoding="utf-8" ?> <Request xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <Insert xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <RECORDSET> <[TABLE_NAME]RECORDINSERT> <[FIELD1_NAME]>[value1]</[FIELD1_NAME]> <[FIELD2_NAME]>[value2]</[FIELD2_NAME]> ... </[TABLE_NAME]RECORDINSERT> </RECORDSET> </Insert> <[SP_NAME] xmlns="[VERSION]/[SCHEMA]/Procedure" /> <Delete xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <FILTER>[WHERE_clause]</FILTER> </Delete> </Request> </pre>
Composite Operation Response	<pre> <?xml version="1.0" encoding="utf-8" ?> <RequestResponse xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <InsertResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <InsertResult>[value]</InsertResult> </InsertResponse> <[SP_NAME]Response xmlns="[VERSION]/[SCHEMA]/Procedure"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]Response> <DeleteResponse xmlns="[VERSION]/[SCHEMA]/Table/[TABLE_NAME]"> <DeleteResult>[value]</DeleteResult> </DeleteResponse> </RequestResponse> </pre>

[VERSION] = The message version string; for example, http://Microsoft.LobServices.OracleDB/2007/03

[PROJECT_NAME] = Name of the BizTalk project that contains the composite operation schema.

[COMPOSITE_SCHEMA_NAME] = Name of the composite operation schema given by the user.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = Name of the table; for example, EMPLOYEE.

[FIELD1_NAME] = Table field name; for example, NAME.

[SP_NAME] = The packaged stored procedure to be executed; for example, ADD_EMP_DETAILS.

[PRM1_NAME] = The name of the Oracle parameter in the stored procedure.

Message Action for the Composite Operation

The message action for the composite operation is "http://Microsoft.LobServices.OracleDB/2007/03/CompositeOperation."

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Message Schemas for the Notification Operation

The Microsoft BizTalk Adapter for Oracle Database surfaces the Notification operation to receive database change notifications from the Oracle database.

You configure the Notification operation by setting binding properties in the Oracle Database adapter. For more information about the Notification-related binding properties, see [Working with BizTalk Adapter for Oracle Database Binding Properties](#). You set the **NotificationStatement** binding property to specify a SELECT statement for the query notification.

Message Structure for the Notification Operation

The following table shows the XML message structure for the Notification operation.

Operation	XML Message	Description
Notification	<pre><?xml version="1.0" encoding="utf-8" ?> <Notification xmlns="http://Microsoft.LobServices.OracleDB/2007/03/Notification"> <Info>Value</Info> <Source>Value</Source> </Notification> <Type>Value</Type> </Notification></pre>	<p>This is the inbound message that is sent by the Oracle database to the adapter clients. In the message:</p> <ul style="list-style-type: none"> • The <code><Info></code> tag indicates the reason for the notification. For example, an "insert" value in this tag indicates that data has been inserted in one or more of the tables referenced in the notification statement. • The <code><Source></code> tag indicates the source for the notification. For example, a "data" value in this tag indicates a change in the data in a referenced object. Similarly, an "object" value in this tag indicates a change in a referenced object. • The <code><Type></code> tag indicates the type of data change. For example, an "Update" value in the <code><Type></code> tag indicates that the results of the query have been updated.

Message Action for the Notification Operation

The message action for the notification operation is "http://Microsoft.LobServices.OracleDB/2007/03/Notification".

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Metadata Node IDs

The Oracle Database adapter surfaces Oracle database artifacts in a hierarchical manner. The following table lists the node types and node IDs for Oracle database artifacts that the Oracle Database adapter surfaces. The node ID is the absolute path of the node that is used in the **IMetadataRetrievalContract Browse**, **Search**, and **GetMetadata** methods.

Artifact Display Name	Node Type	Node ID	Example	Description
--	CATEGORY	/	/	WCF LOB Adapter SDK root node. Returns all first-level nodes ; this includes the SQLEXECUTE operation node, the POLLINGSTMT operation node, and all schema nodes
SQLEXECUTE	OPERATION	[VERSION]/SQLEXECUTE	http://Microsoft.LobServices.OracleDB/2007/03/SQLEXECUTE	SQLEXECUTE operation node. Returns WSDL for the SQLEXECUTE operation.
POLLINGSTMT	OPERATION	[VERSION]/POLLINGSTMT	http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMT	POLLINGSTMT operation node. Returns WSDL for the POLLINGSTMT operation.
[DB_SCHEMA]	CATEGORY	[VERSION]/[DB_SCHEMA]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT	Schema node. Returns general category nodes (Table, View, Procedure, Function, and Package) for the specified schema.
Table	CATEGORY	[VERSION]/[DB_SCHEMA]/Table	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table	Schema tables node. Returns all table nodes for the specified schema.
[DB_TABLE]	CATEGORY	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP	Table node. Returns all operation nodes (Insert, Select, Update, Delete, ReadLOB, and UpdateLOB) for the specified table. (ReadLOB and UpdateLOB are only returned for tables that contain a LOB column.)
Insert	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/Insert	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert	Table Insert operation node. Returns WSDL for the Insert operation for the specified table.
Select	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/Select	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Select	Table Select operation node. Returns WSDL for the Select operation for the specified table.

Update	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/Update	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Update	Table Update operation node. Returns WSDL for the Update operation for the specified table.
Delete	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/Delete	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete	Table Delete operation node. Returns WSDL for the Delete operation for the specified table.
ReadLOB	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/ReadLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/ReadLOB	Table ReadLOB operation node. Returns WSDL for the ReadLOB operation for the specified table. (Only surfaced if the table contains a LOB column.)
UpdateLOB	OPERATION	[VERSION]/[DB_SCHEMA]/Table/[DB_TABLE]/UpdateLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/UpdateLOB	Table UpdateLOB operation node. Returns WSDL for the UpdateLOB operation for the specified table. (Only surfaced if the table contains a LOB column.)
View	CATEGORY	[VERSION]/[DB_SCHEMA]/View	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View	Schema views node. Returns all view nodes for the specified schema.
[DB_VIEW]	CATEGORY	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW	View node. Returns all operation nodes (Insert, Select, Update, Delete, ReadLOB, and UpdateLOB) for the specified view. (ReadLOB and UpdateLOB are only returned for views that contain a LOB column.)
Insert	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/Insert	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/Insert	View Insert operation node. Returns WSDL for the Insert operation for the specified view.
Select	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/Select	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/Select	View Select operation node. Returns WSDL for the Select operation for the specified view.
Update	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/Update	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/Update	View Update operation node. Returns WSDL for the Update operation for the specified view.
Delete	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/Delete	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/Delete	View Delete operation node. Returns WSDL for the Delete operation for the specified view.

ReadLOB	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/ReadLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/ReadLOB	View ReadLOB operation node. Returns WSDL for the ReadLOB operation for the specified view. (Only surfaced if the view contains a LOB column.)
UpdateLOB	OPERATION	[VERSION]/[DB_SCHEMA]/View/[DB_VIEW]/UpdateLOB	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/View/SALES_VIEW/UpdateLOB	View Update operation node. Returns WSDL for the UpdateLOB operation for the specified table. (Only surfaced if the view contains a LOB column.)
Procedure	CATEGORY	[VERSION]/[DB_SCHEMA]/Procedure	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure	Schema procedures node. Returns all procedures for the specified schema.
[DB_PROCEDURE]	OPERATION	[VERSION]/[DB_SCHEMA]/Procedure/[DB_PROCEDURE]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Procedure/SP_GENREPORT	Procedure node. Returns the WSDL for the specified procedure.
Function	CATEGORY	[VERSION]/[DB_SCHEMA]/Function	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Function	Schema functions node. Returns all functions for the specified schema.
[DB_FUNCTION]	OPERATION	[VERSION]/[DB_SCHEMA]/Function/[DB_FUNCTION]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Function/FN_GETUSERID	Function node. Returns the WSDL for the specified function.
Package	CATEGORY	[VERSION]/[DB_SCHEMA]/Package	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package	Schema packages node. Returns all packages for the specified schema.
[DB_PACKAGE]	CATEGORY	[VERSION]/[DB_SCHEMA]/Package/[DB_PACKAGE]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG	Package node. Returns all procedures and functions for the specified package.
[PACKAGE_PROCEDURE]	OPERATION	[VERSION]/[DB_SCHEMA]/Package/[DB_PACKAGE]/[PACKAGE_PROCEDURE]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/GET_ACCOUNT	Package procedure node. Returns the WSDL for the specified package procedure.
[PACKAGE_FUNCTION]	OPERATION	[VERSION]/[DB_SCHEMA]/Package/[DB_PACKAGE]/[PACKAGE_FUNCTION]	http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Package/ACCOUNT_PKG/CREATE_ACCOUNT	Package function node. Returns the WSDL for the specified package function.

[VERSION] = The version string; for example, http://Microsoft.LobServices.OracleDB/2007/03.

[DB_SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[DB_TABLE] = The name of an Oracle table; for example, EMP.

[DB_VIEW] = The name of an Oracle view; for example, SALES_VIEW.

[DB_PROCEDURE] = The name of an Oracle procedure; for example, SP_GENREPORT.

[DB_FUNCTION] = The name of an Oracle function; for example, FN_GETUSERID.

[DB_PACKAGE] = The name of an Oracle package; for example, ACCOUNT_PKG.

[PACK_PROCEDURE] = The name of a package procedure; for example, GET_ACCOUNT.

[PACK_FUNCTION] = The name of a package function; for example, CREATE_ACCOUNT.

Metadata Search and Node IDs

Metadata search is a powerful feature that the WCF LOB Adapter SDK surfaces as part of its **MetadataRetrievalContract** interface. The Oracle Database adapter uses this feature to support searching on the following Oracle artifacts. The metadata search scope is restricted to the level immediately under the node at which the search operation is performed. For example, to search for a function, you must be searching under \[Schema]\Functions. Recursive search is not supported.

Artifact	Node ID	Node Type Returned	Description
[DB_SCHEMA]	/(i.e. Root node)	CATEGORY	Return all schema nodes that match the search expression.
[DB_TABLE]	/[VERSION]/[DB_SCHEMA]/Table	CATEGORY	Return all table nodes in the specified schema that match the search expression.
[DB_VIEW]	/[VERSION]/[DB_SCHEMA]/View	CATEGORY	Return all view nodes in the specified schema that match the search expression.
[DB_PROCEDURE]	/[VERSION]/[DB_SCHEMA]/Procedure	OPERATION	Return all procedure nodes in the specified schema that match the search expression.
[DB_FUNCTION]	/[VERSION]/[DB_SCHEMA]/Function	OPERATION	Return all function nodes in the specified schema that match the search expression.
[DB_PACKAGE]	/[VERSION]/[DB_SCHEMA]/Package	CATEGORY	Return all package nodes (category) in the specified schema that match the search expression.
[PACK_PROCEDURE] and [PACK_FUNCTION]	/[VERSION]/[DB_SCHEMA]/Package/[DB_PACKAGE]	OPERATION	Return all function and procedure nodes (operation) in the specified package that match the search expression.

[VERSION] = The version string; for example, http://Microsoft.LobServices/2007/03.

[DB_SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[DB_TABLE] = The name of an Oracle table; for example, EMP.

[DB_VIEW] = The name of an Oracle view; for example, SALES_VIEW.

[DB_PROCEDURE] = The name of an Oracle procedure; for example, SP_GENREPORT.

[DB_FUNCTION] = The name of an Oracle function; for example, FN_GETUSERID.

[DB_PACKAGE] = The name of an Oracle package; for example, ACCOUNT_PKG.

[PACK_PROCEDURE] = The name of a package procedure; for example, GET_ACCOUNT.

[PACK_FUNCTION] = The name of a package function; for example, CREATE_ACCOUNT.

You can specify search expressions that are compatible with any valid expression that can be used for the Oracle LIKE operator. For example, to perform a search on the tables contained in a schema, the Oracle Database adapter executes the following SQL:
`SELECT TABLE_NAME FROM ALL_TABLES WHERE OWNER = '[OWNER_NAME]' AND TABLE_NAME LIKE '[SEARCH_STR]'.`

The following table lists the special characters that the Oracle Database adapter supports in search expressions.

Special Character	Interpretation
% (percentage)	Matches zero or more characters; for example, "A%" matches "A", "AB", "ABC", and so on.
_ (underscore)	Matches exactly 1 character; for example, "A_" matches "AB", "AC", "AD", and so on.
\ (escape)	Escapes the special meaning of '%' and '_'; for example, "A_B" matches "A_B".

Metadata Retrieval and Node IDs

The following table summarizes the metadata characteristics returned by Oracle Database adapter.

Artifact	Metadata Characteristics
Table or View	<ul style="list-style-type: none"> • Table name. • Table field names. • Table field data types are mapped to simple or complex WSDL types. • Table field length is mapped to facet maxLength. • Table field primary key constraint is mapped to facet minOccurs = 1. • Table field NULL constraint is mapped to facet isNillable = true. • Table operations <ul style="list-style-type: none"> • INSERT • SELECT • UPDATE • DELETE • READLOB (if the table contains Oracle LOB type field) • UPDATELOB (if the table contains Oracle LOB type field)
Procedure or Function	<ul style="list-style-type: none"> • Procedure or function name is mapped to the operation name. • Procedure or function parameter names. • Procedure or function parameter data types are mapped to WSDL types. • Procedure or function parameter direction is mapped to WSDL parameter direction. • Procedure parameter or function parameter data type length is mapped to facet maxLength. • Procedure or function parameter order is mapped to element sequence. • Function return data type is mapped to WSDL type. • Function return data type length is mapped to facet maxLength.

Package Procedure or Function.	<ul style="list-style-type: none">• Package name.• Other procedure and function characteristics as listed above.
--------------------------------	---

For detailed information about the format of the metadata that the Oracle Database adapter exposes for specific artifacts and operations on the Oracle database, see [Messages and Message Schemas for BizTalk Adapter for Oracle Database](#).

See Also

Other Resources

[Technical Reference](#)

Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Oracle Database

You can use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class or a WCF service contract (interface) for operations that the Microsoft BizTalk Adapter for Oracle Database exposes. After you run svcutil.exe to generate either a WCF client class or a WCF service contract, you can include the generated file in your code and create instances of the generated class or implement a WCF service from the contract to perform operations on the Oracle database.

Using svcutil.exe requires you to supply a connection URI that contains credentials. Because, by default, the Oracle Database adapter disables credentials in the connection URI, you must configure svcutil.exe to use a non-default binding for the Oracle Database adapter.

The following sections show you how to configure svcutil.exe and how to use svcutil.exe to generate WCF client code or a WCF service contract with the Oracle Database adapter.

Configuring svcutil.exe for the Oracle Database Adapter

To configure svcutil.exe to use a non-default binding, you must create a local copy of svcutil.exe and then create or modify a local copy of the svcutil.exe.config configuration file.

To configure svcutil.exe for a non-default binding

1. Create a folder, and copy svcutil.exe into the new folder. You can typically find svcutil.exe at the Windows SDK installation location, specifically, C:\Program Files\Microsoft SDKs\Windows\v6.0\Bin.
2. Create a file named svcutil.exe.config in the new folder.
3. Add a binding and a client endpoint to the svcutil.exe.config file. You must run svcutil.exe from the new folder to ensure that the correct configuration is used.

Important

The name attribute of the client endpoint must specify the scheme used in the connection URI. This value is case-sensitive.

```
<configuration>
  <system.serviceModel>
    <client>
      <!-- the name should match the required scheme of the WS-Metadata Exchange endpoint
      and the contract should be "IMetadataExchange" -->
      <endpoint name="oracledb"
        binding="oracleDBBinding"
        bindingConfiguration="OracleDBBinding"
        contract="IMetadataExchange" />
    </client>
    <bindings>
      <oracleDBBinding>
        <binding name="OracleDBBinding" acceptCredentialsInUri="true" />
      </oracleDBBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

Note

You can set any of the binding properties of the Oracle Database adapter in the binding configuration.

For more information about configuring a non-default binding for svcutil.exe, see the "Custom Secure Metadata Endpoint" topic in the WCF documentation at <http://go.microsoft.com/fwlink/?LinkId=96077>.

Configuring a Non-Default Binding for the POLLINGSTMT Operation

To use svcutil.exe to create a WCF service contract for the POLLINGSTMT operation, you must configure the non-default binding to include the **pollingStatement** property, in addition to **acceptCredentialsInUri**. The **pollingStatement** must contain the SELECT statement that targets the table. The Oracle Database adapter uses this property to generate the class that represents the strongly-typed result set that the POLLINGSTMT operation returns. The following example shows a binding configuration that is used to generate a WCF service contract for a POLLINGSTMT operation that targets the /SCOTT/EMP table.

```
<bindings>
  <oracleDBBinding>
    <binding name="OracleDBBinding" acceptCredentialsInUri="true"
      pollingStatement="SELECT * FROM EMP FOR UPDATE" />
  </oracleDBBinding>
</bindings>
```

Creating a WCF Client Class or WCF Service Contract with svcutil.exe

To use svcutil.exe to generate WCF client code or a WCF service contract (interface) for the Oracle Database adapter, you must supply a connection URI that specifies an WS-Metadata Exchange (MEX) endpoint and the operation or operations for which you want svcutil.exe to generate code. You must also specify connection credentials for the Oracle database in the connection URI.

Note

Before you can use svcutil.exe with the Oracle Database adapter, you must configure it to use a non-default binding; for information about how to do this, see [Configuring svcutil.exe for the Oracle Database Adapter](#).

You specify a MEX endpoint and target operations in the Oracle Database adapter connection URI in the following manner:

- You must include the "wsdl" parameter in the query_string. If it is the first parameter in the query_string, it is specified just after the question mark (?). If it is not the first parameter, it should be preceded with an ampersand (&).

- You must follow the "wsdl" parameter by one or more "op" parameters. Each "op" parameter is preceded by an ampersand (&) and specifies the node ID of a target operation.

The following three examples show how to target various operations by using svcutil.exe.

This example creates a WCF client class for an Insert operation on the /SCOTT/EMP table.

```
.\svcutil "oracledb://User=SCOTT;Password=TIGER@ADAPTER?wsdl&op=http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert"
```

This example creates a WCF client class for the Insert and the Delete operations on the /SCOTT/EMP table.

```
.\svcutil "oracledb://User=SCOTT;Password=TIGER@ADAPTER?  
wsdl&op=http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Insert&op=http://Microsoft.LobServices.OracleDB/2007/03/SCOTT/Table/EMP/Delete"
```

This example creates a WCF service contract for the POLLINGSTMT operation. (To use svcutil.exe to generate a WCF service contract for the POLLINGSTMT operation, you must configure a non-default binding for svcutil.exe that includes a polling statement.)

```
.\svcutil "oracledb://User=SCOTT;Password=TIGER@ADAPTER?wsdl&op=http://Microsoft.LobServices.OracleDB/2007/03/POLLINGSTMT"
```

◆ Important

You must place the connection URI in quotation marks on the command line. Otherwise, svcutil.exe attempts to retrieve metadata for operations that the Oracle Database adapter does not support. The results of such an attempt are undefined.

By default, svcutil.exe places the generated code in the output.cs file; however, you can change the name of the output file and many other options that svcutil.exe uses by setting command-line switches. For more information about the options that svcutil.exe supports, see the "ServiceModel Metadata Utility Tool (Svcutil.exe)" topic in the WCF documentation at <http://go.microsoft.com/fwlink/?LinkId=72777>.

Svcutil.exe does not provide the capability to search for operations (for example, by using wildcard characters). You must explicitly specify node IDs for the specific operations you want to target. You cannot specify node IDs that refer only to categories. For more information about the node IDs that the Oracle Database adapter surfaces, see [Metadata Node IDs](#).

The Add Adapter Service Reference Visual Studio Plug-in provides advanced browse and search capabilities that can greatly simplify generating a WCF client class and WCF service contract. For more information about the Add Adapter Service Reference Plug-in, see [Generating a WCF Client or a WCF Service Contract for Oracle Database Artifacts](#).

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle Database](#)

Glossary

The following terms and definitions are used in Microsoft BizTalk Adapter for Oracle Database Help.

Microsoft BizTalk Adapter Pack Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

adapter	A WCF-based component that helps exchange messages between applications (for example, a line-of-business system) and BizTalk Server. The adapter consists of design-time components and run-time components for receive and send operations.
adapter client	An application that interacts with a line-of-business (LOB) system through the adapter.

B

BFILE	An Oracle data type that enables access to binary file LOBs that are stored in file systems external to the Oracle database. A BFILE column stores a BFILE locator, which represents the directory name and file name that contains the data on the server file system.
binary large object	<ol style="list-style-type: none"> 1. A large piece of data, such as a bitmap, characterized by large field values, an unpredictable table size, and data that is formless from the perspective of an application. 2. A keyword that designates the BLOB structure that contains information about a block of data.
binding	A process by which software components and layers are linked together. When a network component is installed, the binding relationships and dependencies for the components are established. Binding allows components to communicate with each other. In BizTalk Server, an established mapping between an orchestration adapter-agnostic endpoint (port or role link) and physical adapter-specific endpoints (send/receive ports or party).
BizTalk Server	Connects diverse software. BizTalk Server enables you to create and modify process logic that uses that software. BizTalk Server also enables information workers to monitor running processes, interact with trading partners, and perform other business-oriented tasks.

C

channel	A concrete implementation of a binding element. The binding represents the configuration, and the channel is the implementation associated with that configuration. Therefore, there is a channel associated with each binding element. Channels stack on top of each other to create the concrete implementation of the binding: the channel stack.
connection URI	A string that identifies a resource in a distributed environment. Adapters use a connection Uniform Resource Identifier (URI) that contains the information necessary to establish a connection with the LOB system.
contract	Specifies the collection and structure of messages required to access the operations offered by the service.

D

data manipulation language (DML)	The subset of SQL statements that is used to retrieve and manipulate data. DML statements typically start with SELECT, INSERT, UPDATE, or DELETE.
design-time experience	Procedures and operations that a developer performs during design time; for example, using the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas.

E

endpoint address	A network address that identifies the location of a Windows Communication Foundation (WCF) service endpoint. For an adapter, the endpoint address is expressed as a connection Uniform Resource Identifier (URI) that contains location and connection parameters. The adapter can use these to establish a connection to the underlying line-of-business (LOB) system.
Enterprise Single Sign-on system	An SSO database, a master secret server, and one or more Enterprise Single Sign-On (SSO) servers. These servers define the mapping between the Windows and non-Windows credentials, look up the credentials in the SSO database, and are used for administering the SSO system. The SSO database is also used as a configuration store to hold custom configuration data for adapters.
Extensible Markup Language	A markup language designed to describe data. XML tags are not predefined.

G

GAC	See global assembly cache.
global assembly cache (GAC)	A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer. Applications deployed in the global assembly cache must have a strong name.

H

Health and Activity Tracking (HAT)	A user interface that enables specified data to be tracked. This interface can be used to monitor the health of the BizTalk Server farm and to track and view specific events and messages.
------------------------------------	---

I

inbound operation	An operation that is invoked by a line-of-business (LOB) system on the adapter.
-------------------	---

L

local naming method	The Oracle naming method that is supported by the Oracle Database adapter. In this naming method, the Oracle client resolves the net service name to an entry in the local tnsnames.ora file.
---------------------	---

M

metadata	In WCF, refers to a description of the contract exposed by a service. This is known as the service description and is expressed in a WSDL document. The metadata exposed by an adapter describes the (interface to) the operations that it can perform on the underlying LOB system.
Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK	The specifications for building BizTalk adapters using open standards based on Web services.

N

naming method	Oracle naming methods determine how the Oracle client obtains connection information for an Oracle database service (instance). You can configure the Oracle client to use specific naming methods by using the Oracle Net Configuration Assistant. The Oracle Database adapter supports the Local Naming method.
net service name	An alias used by the Oracle client to obtain connection information for the Oracle database. You supply a net service name as one of the connection properties in the connection URI.

O

one-way	A message exchange pattern (MEP) in which the sender sends a message, but no response is returned by the receiver. In BizTalk Server, MEPs are referred to as communication patterns.
outbound operation	An operation that is invoked by the adapter on the line-of-business system (LOB).

output.cs	The default output file created by the ServiceModel Metadata Utility tool (svcutil.exe).
-----------	--

P

polling	A technique that device drivers use to find out from multiple devices whether they contain data to transmit. The devices are polled one at a time.
proxy	In WCF, refers to a managed-code object that implements the service contract exposed by a service. The WCF service model is based on the use of such proxies. In the WCF service model, the service contract is expressed as a .NET interface.

R

REF CURSOR	An Oracle PL/SQL data type that represents a pointer to a result set in the Oracle database. A REF CURSOR type enables input and output streaming of data, and is ideal for transferring large amounts of data to and from a PL/SQL code block.
request-reply	A message exchange pattern (MEP) in which the sender sends a request message and expects a response message from the receiver. In BizTalk Server, MEPs are referred to as communication patterns. Depending on the messaging technology and the direction of the request message (inbound or outbound), this pattern is also called request-reply or solicit-response.
runtime experience	Procedures and operations performed by a developer during run time or when deploying a solution; for example, creating a physical port binding from the BizTalk Server Administration console.

S

schema	The structure for a message. A schema can contain multiple subschema.
ServiceModel Metadata Utility Tool (svcutil.exe)	A command-line utility that is included with WCF. It is used to create service model proxy code from the service description (metadata) that is exposed by a WCF service such as an adapter. For outbound operations, the tool creates a WCF client class and helper code; for inbound operations, the tool creates a WCF service contract and helper code.
Simple Object Access Protocol	See SOAP.
SOAP	A simple, XML-based protocol for exchanging structured and type information in decentralized, distributed environments. WCF is based on the exchange of SOAP messages between clients and services to invoke operations and return results.
SOAP message	A well-formed XML document. It should use the SOAP envelope and SOAP encoding namespaces and include an optional XML declaration, followed by a SOAP envelope (the root element), which is made up of an optional SOAP header and a SOAP message body.
SQL Server Integration Services	A component that is used to import, export, and transform data from different data sources. Previously called data transformation service (DTS).
SSIS	See SQL Server Integration Services.
SSO	See Enterprise Single Sign-on system.
strongly-typed data	A data set or result set that is bound to an underlying object type. Each row in a strongly-typed XML data set is composed of typed, named elements that correspond to fields of the underlying object type.

svcutil.exe	See ServiceModel Metadata Utility Tool.
-------------	---

U

Uniform Resource Identifier (URI)	See connection URI
-----------------------------------	--------------------

W

WCF	See Windows Communication Foundation
WCF channel model	A programming model that relies on several interfaces and other types. Channels provide a low-level programming model for sending and receiving messages.
WCF client	A client-application construct that exposes the service operations as methods. You can use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool to generate a WCF client class from the metadata exposed by an adapter.
WCF LOB Adapter SDK	See Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
WCF service contract	A managed-code representation of the service contract. It is expressed as an interface in which classes and methods are attributed to define the service, operation, message, and data contracts used to communicate with a service. You can use the ServiceModel Metadata Utility tool or the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF service contract from the metadata exposed by an adapter. You implement the WCF service contract to receive operations from an LOB system.
WCF service model	A WCF programming model in which a service is represented as a managed code object. The operations exposed by the service are represented as methods with strongly-typed data.
weakly-typed data	A data set or result set that is not bound to an underlying object type. Each row in a weakly-typed XML data set is composed of a collection of generic columns in which attributes describe the name and type of each element.
Web services	A unit of application logic providing data and services to other applications. Applications access XML Web services using standard Web protocols and data formats such as HTTP, XML, and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web , and are a cornerstone of the Microsoft .NET programming model.
Web Services Description Language (WSDL)	An XML-based language that describes a service as a set of endpoints that operate on messages. The WSDL document describes the service contract, operation contracts, message contracts, and data contracts that a client must use to interface with the service.
Windows Communication Foundation (WCF)	A Microsoft service-oriented communication infrastructure. The framework inherently provides clients with a service programming model and a channel programming model for finer control of message exchanges.
WSDL	See Web Services Description Language.
WS-Metadata Exchange (MX) endpoint	An endpoint exposed by a WCF service, such as an adapter, that implements the IMetadataExchange interface. A WS-Metadata Exchange endpoint can be used to retrieve a service description (WSDL) for operations exposed by an adapter on the target system.

X

XML	See Extensible Markup Language.
XML Schema definition language (XSD)	A schema language. An XML Schema defines the elements, attributes, and data types that comply with the World Wide Web Consortium (W3C) XML Schema Part 1: Structures Recommendation for the XML Schema Definition Language. The W3C XML Schema Part 2: Datatypes Recommendation is the recommendation for defining data types that are used in XML schemas. The XML Schema definition language enables you to define the structure and data types for XML messages.
XSD	See XML Schema definition language.

Class Library

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Namespaces

Namespace
Microsoft.Adapters.OracleDB

Microsoft.Adapters.OracleDB Namespace

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Classes

Class	Description
OracleDBAdapter	Creates an adapter for Oracle DB (implemented as a transport binding element).
OracleDBAdapterBindingSection	Represents a configuration section for Oracle DB adapter binding.
OracleDBBinding	Creates a binding for Oracle DB adapter.
OracleDBBindingConfigurationElement	Represents the configuration element for Oracle DB adapter binding.
OracleDBConnectionUri	Represents the class for building the OracleDBConnectionUri.
OracleDBInboundTransactionBehavior	Represents the behavior that controls the transaction settings for WCF-Dispatcher initiated transactions.
OracleDBPerformanceCounterConstants	Specifies a class exposing constants pertaining to OracleDB Adapter.

Enumerations

Enumeration	Description
InboundOperation	Specifies the Inbound Operation to be performed.
OracleServiceType	Specifies the type of Oracle database service.
TransactionIsolationLevel	Specifies the isolation level of the Transaction.

InboundOperation Enumeration

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Specifies the Inbound Operation to be performed.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Enumeration InboundOperation
```

C#

```
public enum InboundOperation
```

C++

```
public enum class InboundOperation
```

J#

```
public enum InboundOperation
```

JScript

```
public enum InboundOperation
```

Members

Member name	Description
Notification	The Notification Inbound Operation.
Polling	The Polling Inbound Operation.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Creates an adapter for Oracle DB (implemented as a transport binding element).

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class OracleDBAdapter
    Inherits CommonAdapter
```

C#

```
public sealed class OracleDBAdapter : CommonAdapter
```

C++

```
public ref class OracleDBAdapter sealed : public CommonAdapter
```

J#

```
public final class OracleDBAdapter extends CommonAdapter
```

JScript

```
public final class OracleDBAdapter extends CommonAdapter
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.BindingElement](#)

[System.ServiceModel.Channels.TransportBindingElement](#)

[Microsoft.ServiceModel.Channels.Common.Adapter](#)

[Microsoft.Adapters.AdapterUtilities.CommonAdapter](#)

Microsoft.Adapters.OracleDB.OracleDBAdapter

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Creates an adapter for Oracle DB (implemented as a transport binding element).

The following tables list the members exposed by the [OracleDBAdapter](#) type.

Public Constructors

Name	Description
OracleDBAdapter	Overloaded. Initializes a new instance of the OracleDBAdapter class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 EnableBizTalkCompatibilityMode	Gets or sets whether the value of this property is enabled when using the adapter with Biztalk Server.
 EnablePerformanceCounters	Gets or sets whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets whether safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the type of inbound operation to be performed.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for record insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size of a long data type column. Obsolete.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 MetadataPooling	Gets or sets whether metadata pooling is used.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 NotificationPort	Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the transacted polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried. Obsolete.
 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the post poll statement.
 Scheme	Overridden. Gets the scheme name for the specified object.

 SkipNilNodes	Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets whether statement cache is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached.
 TransactedReceiveEnabled	(inherited from Adapter)
 TransactionIsolationLevel	Gets or sets the Oracle transaction isolation level. Obsolete.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether oracle connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Protected Properties

	Name	Description
	Namespace	(inherited from Adapter)
	Settings	(inherited from Adapter)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	BuildConnectionFactory	(inherited from Adapter)
	BuildConnectionUri	(inherited from Adapter)
	BuildWsdlRetrieval	(inherited from Adapter)
	CloneAdapter	(inherited from Adapter)
	Finalize	(inherited from Object)
	IsHandlerSupported	(inherited from Adapter)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[OracleDBAdapter Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBAdapter](#) class.

Overload List

Name	Description
OracleDBAdapter ()	Initializes a new instance of the Oracle DB adapter class.
OracleDBAdapter (OracleDBAdapter)	Initializes a new instance of the Oracle DB adapter class with a binding.

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Constructor ()

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the Oracle DB adapter class.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleDBAdapter ()
```

C++

```
public:  
OracleDBAdapter ()
```

J#

```
public OracleDBAdapter ()
```

JScript

```
public function OracleDBAdapter ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Constructor (OracleDBAdapter)

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the Oracle DB adapter class with a binding.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    other As OracleDBAdapter _  
)
```

C#

```
public OracleDBAdapter (  
    OracleDBAdapter other  
)
```

C++

```
public:  
OracleDBAdapter (  
    OracleDBAdapter^ other  
)
```

J#

```
public OracleDBAdapter (  
    OracleDBAdapter other  
)
```

JScript

```
public function OracleDBAdapter (  
    other : OracleDBAdapter  
)
```

Parameters

other

The other Oracle DB adapter.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	BuildConnectionFactory	(inherited from Adapter)
	BuildConnectionUri	(inherited from Adapter)
	BuildWsdRetrieval	(inherited from Adapter)
	CloneAdapter	(inherited from Adapter)
	Finalize	(inherited from Object)
	IsHandlerSupported	(inherited from Adapter)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleDBAdapter Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 EnableBizTalkCompatibilityMode	Gets or sets whether the value of this property is enabled when using the adapter with Biztalk Server.
 EnablePerformanceCounters	Gets or sets whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets whether safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the type of inbound operation to be performed.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for record insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size of a long data type column. Obsolete.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 MetadataPooling	Gets or sets whether metadata pooling is used.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 NotificationPort	Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the transacted polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried. Obsolete.
 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the post poll statement.
 Scheme	Overridden. Gets the scheme name for the specified object.
 SkipNilNodes	Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets whether statement cache is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached.
 TransactedReceiveEnabled	(inherited from Adapter)

 TransactionIsolationLevel	Gets or sets the Oracle transaction isolation level. Obsolete.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether oracle connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Protected Properties

	Name	Description
	Namespace	(inherited from Adapter)
	Settings	(inherited from Adapter)

See Also

Reference

[OracleDBAdapter Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.AcceptCredentialsInUri Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AcceptCredentialsInUri () : boolean

CompilerGeneratedAttribute
public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle connection URI can contain user credentials for the Oracle database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.ConnectionLifetime Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property ConnectionLifetime As Integer
```

C#

```
public int ConnectionLifetime { get; set; }
```

C++

```
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int
public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.DataFetchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the amount of data to fetch.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Property DataFetchSize As Long
```

C#

```
public long DataFetchSize { get; set; }
```

C++

```
public:  
virtual property long long DataFetchSize {  
    long long get () sealed;  
    void set (long long value) sealed;  
}
```

J#

```
/** @property */  
public final long get_DataFetchSize ()  
  
/** @property */  
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long  
public final function set DataFetchSize (value : long)
```

Property Value

The data type.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.DecrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Property DecrPoolSize As Integer
```

C#

```
public int DecrPoolSize { get; set; }
```

C++

```
public:  
virtual property int DecrPoolSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_DecrPoolSize ()  
  
/** @property */  
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int  
public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.EnableBizTalkCompatibilityMode Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the value of this property is enabled when using the adapter with Biztalk Server.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
public:  
virtual property bool EnableBizTalkCompatibilityMode {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnableBizTalkCompatibilityMode ()  
  
/** @property */  
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean  
  
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the value of this property is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.EnablePerformanceCounters Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the performance counters are enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property EnablePerformanceCounters As Boolean
```

C#

```
public bool EnablePerformanceCounters { get; set; }
```

C++

```
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if performance counters are enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.EnableSafeTyping Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether safe typing is enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property EnableSafeTyping As Boolean
```

C#

```
public bool EnableSafeTyping { get; set; }
```

C++

```
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableSafeTyping ()

/** @property */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
public final function get EnableSafeTyping () : boolean

public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if safe typing is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.GeneratedUserTypesAssemblyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
public:  
virtual property String^ GeneratedUserTypesAssemblyFilePath {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

J#

```
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final String get_GeneratedUserTypesAssemblyFilePath ()  
  
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute  
public final function get GeneratedUserTypesAssemblyFilePath () : String  
  
CompilerGeneratedAttribute  
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.GeneratedUserTypesAssemblyKeyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
public:  
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

J#

```
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()  
  
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute  
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String  
  
CompilerGeneratedAttribute  
public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.InboundOperationType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the type of inbound operation to be performed.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property InboundOperationType As InboundOperation
```

C#

```
public InboundOperation InboundOperationType { get; set; }
```

C++

```
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The inbound operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.IncrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property IncrPoolSize As Integer
```

C#

```
public int IncrPoolSize { get; set; }
```

C++

```
public:  
virtual property int IncrPoolSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_IncrPoolSize ()  
  
/** @property */  
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int  
  
public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections to be created.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.InsertBatchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the batch size for record insert operations.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property InsertBatchSize As Integer
```

C#

```
public int InsertBatchSize { get; set; }
```

C++

```
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int

public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for record insert operations.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.LongDatatypeColumnSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the maximum size of a long data type column.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")> _
Public Property LongDatatypeColumnSize As Long
```

C#

```
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")]
public long LongDatatypeColumnSize { get; set; }
```

C++

```
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation
for suggested alternatives")]
public:
virtual property long long LongDatatypeColumnSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (long value)
```

JScript

```
public final function get LongDatatypeColumnSize () : long
public final function set LongDatatypeColumnSize (value : long)
```

Property Value

The maximum size of a long data type column.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.MaxOutputAssociativeArrayElements Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
public:  
virtual property int MaxOutputAssociativeArrayElements {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_MaxOutputAssociativeArrayElements ()  
  
/** @property */  
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get_MaxOutputAssociativeArrayElements () : int  
  
public final function set_MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array that the binding creates.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.MaxPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property MaxPoolSize As Integer
```

C#

```
public int MaxPoolSize { get; set; }
```

C++

```
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int
public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.MetadataPooling Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether metadata pooling is used.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public Property MetadataPooling As Boolean
```

C#

```
public bool MetadataPooling { get; set; }
```

C++

```
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_MetadataPooling ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MetadataPooling (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MetadataPooling () : boolean

CompilerGeneratedAttribute
public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if metadata pooling is used; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.MinPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the minimum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property MinPoolSize As Integer
```

C#

```
public int MinPoolSize { get; set; }
```

C++

```
public:  
virtual property int MinPoolSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_MinPoolSize ()  
  
/** @property */  
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int  
  
public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.NotificationPort Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property NotificationPort As Integer
```

C#

```
public int NotificationPort { get; set; }
```

C++

```
public:  
virtual property int NotificationPort {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_NotificationPort ()  
  
/** @property */  
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int  
  
public final function set NotificationPort (value : int)
```

Property Value

The notification port number.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.NotificationStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property NotificationStatement As String
```

C#

```
public string NotificationStatement { get; set; }
```

C++

```
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

The notification statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.NotifyOnListenerStart Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property NotifyOnListenerStart As Boolean
```

C#

```
public bool NotifyOnListenerStart { get; set; }
```

C++

```
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the adapter sends a notification message when the listener starts; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PolledDataAvailableStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property PolledDataAvailableStatement As String
```

C#

```
public string PolledDataAvailableStatement { get; set; }
```

C++

```
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The polled data available statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PollingAction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property PollingAction As String
```

C#

```
public string PollingAction { get; set; }
```

C++

```
public:
virtual property String^ PollingAction {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingAction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingAction (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingAction () : String

CompilerGeneratedAttribute
public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PollingInterval Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the transacted polling interval.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property PollingInterval As Integer
```

C#

```
public int PollingInterval { get; set; }
```

C++

```
public:  
virtual property int PollingInterval {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_PollingInterval ()  
  
/** @property */  
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int  
  
public final function set PollingInterval (value : int)
```

Property Value

The polling interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PollingRetryCount Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the number of times that polling is retried.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
Public Property PollingRetryCount As Integer
```

C#

```
[BrowsableAttribute(false)]
[ObsoleteAttribute("This field/property/method is deprecated")]
public int PollingRetryCount { get; set; }
```

C++

```
[BrowsableAttribute(false)]
[ObsoleteAttribute(L"This field/property/method is deprecated")]
public:
property int PollingRetryCount {
    int get ();
    void set (int value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public int get_PollingRetryCount ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_PollingRetryCount (int value)
```

JScript

```
CompilerGeneratedAttribute
public function get PollingRetryCount () : int

CompilerGeneratedAttribute
public function set PollingRetryCount (value : int)
```

Property Value

The number of times that polling is retried.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PollingStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the polling statement.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property PollingStatement As String
```

C#

```
public string PollingStatement { get; set; }
```

C++

```
public:
virtual property String^ PollingStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingStatement () : String

CompilerGeneratedAttribute
public final function set PollingStatement (value : String)
```

Property Value

The polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PollWhileDataFound Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the specified object continuously poll data when it is available.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public Property PollWhileDataFound As Boolean
```

C#

```
public bool PollWhileDataFound { get; set; }
```

C++

```
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object continuously poll data when it is available; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.PostPollStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the post poll statement.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property PostPollStatement As String
```

C#

```
public string PostPollStatement { get; set; }
```

C++

```
public:
virtual property String^ PostPollStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PostPollStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PostPollStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PostPollStatement () : String

CompilerGeneratedAttribute
public final function set PostPollStatement (value : String)
```

Property Value

The post poll statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.Scheme Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets the scheme name for the specified object.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property Scheme As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override string Scheme { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property String^ Scheme {
String^ get () override;
}
```

J#

```
/** @property */
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Property Value

The scheme name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.SkipNilNodes Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property SkipNilNodes As Boolean
```

C#

```
public bool SkipNilNodes { get; set; }
```

C++

```
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_SkipNilNodes ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get SkipNilNodes () : boolean

CompilerGeneratedAttribute
public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if inserting nil nodes is skipped; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.StatementCachePurge Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether statement cache is purged.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public Property StatementCachePurge As Boolean
```

C#

```
public bool StatementCachePurge { get; set; }
```

C++

```
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_StatementCachePurge ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get StatementCachePurge () : boolean

CompilerGeneratedAttribute
public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if statement cache is purged; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.StatementCacheSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum number of statements that can be cached.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property StatementCacheSize As Integer
```

C#

```
public int StatementCacheSize { get; set; }
```

C++

```
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int
public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.TransactionIsolationLevel Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the Oracle transaction isolation level.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
Public Property TransactionIsolationLevel As TransactionIsolationLevel
```

C#

```
[BrowsableAttribute(false)]
[ObsoleteAttribute("This field/property/method is deprecated")]
public TransactionIsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[BrowsableAttribute(false)]
[ObsoleteAttribute(L"This field/property/method is deprecated")]
public:
property TransactionIsolationLevel TransactionIsolationLevel {
    TransactionIsolationLevel get ();
    void set (TransactionIsolationLevel value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public TransactionIsolationLevel get_TransactionIsolationLevel ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_TransactionIsolationLevel (TransactionIsolationLevel value)
```

JScript

```
CompilerGeneratedAttribute
public function get TransactionIsolationLevel () : TransactionIsolationLevel

CompilerGeneratedAttribute
public function set TransactionIsolationLevel (value : TransactionIsolationLevel)
```

Property Value

The Oracle transaction isolation level.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.UseAmbientTransaction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Property UseAmbientTransaction As Boolean
```

C#

```
public bool UseAmbientTransaction { get; set; }
```

C++

```
public:  
virtual property bool UseAmbientTransaction {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_UseAmbientTransaction ()  
  
/** @property */  
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean  
  
public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter performs the operations using the transaction context provided by the caller; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.UseOracleConnectionPool Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether oracle connection pool is used.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property UseOracleConnectionPool As Boolean
```

C#

```
public bool UseOracleConnectionPool { get; set; }
```

C++

```
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseOracleConnectionPool ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseOracleConnectionPool () : boolean

CompilerGeneratedAttribute
public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if oracle connection pool is used; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.UserAssembliesLoadPath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property UserAssembliesLoadPath As String
```

C#

```
public string UserAssembliesLoadPath { get; set; }
```

C++

```
public:
virtual property String^ UserAssembliesLoadPath {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_UserAssembliesLoadPath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UserAssembliesLoadPath () : String

CompilerGeneratedAttribute
public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The name of the DLLs.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapter.UseSchemaInNameSpace Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseSchemaInNameSpace () : boolean

CompilerGeneratedAttribute
public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the schema name is included in the xml namespace for operations and their associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapter Class](#)

[OracleDBAdapter Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapterBindingSection Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents a configuration section for Oracle DB adapter binding.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
Public NotInheritable Class OracleDBAdapterBindingSection
    Inherits StandardBindingCollectionElement(Of OracleDBBinding, OracleDBBindingConfigurationElement)
```

C#

```
public sealed class OracleDBAdapterBindingSection : StandardBindingCollectionElement<OracleDBBinding, OracleDBBindingConfigurationElement>
```

C++

```
public ref class OracleDBAdapterBindingSection sealed : public StandardBindingCollectionElement<OracleDBBinding^, OracleDBBindingConfigurationElement^>
```

J#

```
public final class OracleDBAdapterBindingSection extends StandardBindingCollectionElement<OracleDBBinding, OracleDBBindingConfigurationElement>
```

JScript

```
public final class OracleDBAdapterBindingSection extends StandardBindingCollectionElement<OracleDBBinding, OracleDBBindingConfigurationElement>
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.BindingCollectionElement](#)

[System.ServiceModel.Configuration.StandardBindingCollectionElement](#)

Microsoft.Adapters.OracleDB.OracleDBAdapterBindingSection

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapterBindingSection Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapterBindingSection Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents a configuration section for Oracle DB adapter binding.

The following tables list the members exposed by the [OracleDBAdapterBindingSection](#) type.

Public Constructors

Name	Description
OracleDBAdapterBindingSection	Initializes a new instance of the OracleDBAdapterBindingSection class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BindingName	(inherited from BindingCollectionElement)
 Bindings	(inherited from StandardBindingCollectionElement)
 BindingType	(inherited from StandardBindingCollectionElement)
 ConfiguredBindings	(inherited from StandardBindingCollectionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from StandardBindingCollectionElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 ContainsKey	(inherited from StandardBindingCollectionElement)
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 GetDefault	(inherited from StandardBindingCollectionElement)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)

 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ConfigurationElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 TryAdd	(inherited from StandardBindingCollectionElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[OracleDBAdapterBindingSection Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapterBindingSection Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBAdapterBindingSection](#) class.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleDBAdapterBindingSection ()
```

C++

```
public:  
OracleDBAdapterBindingSection ()
```

J#

```
public OracleDBAdapterBindingSection ()
```

JScript

```
public function OracleDBAdapterBindingSection ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBAdapterBindingSection Class](#)

[OracleDBAdapterBindingSection Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapterBindingSection Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	ContainsKey	(inherited from StandardBindingCollectionElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	GetDefault	(inherited from StandardBindingCollectionElement)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ConfigurationElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	TryAdd	(inherited from StandardBindingCollectionElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[OracleDBAdapterBindingSection Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBAdapterBindingSection Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties (see also [Protected Properties](#))

	Name	Description
	BindingName	(inherited from BindingCollectionElement)
	Bindings	(inherited from StandardBindingCollectionElement)
	BindingType	(inherited from StandardBindingCollectionElement)
	ConfiguredBindings	(inherited from StandardBindingCollectionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingCollectionElement)

See Also

Reference

[OracleDBAdapterBindingSection Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Creates a binding for Oracle DB adapter.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class OracleDBBinding
    Inherits AdapterBinding
```

C#

```
public sealed class OracleDBBinding : AdapterBinding
```

C++

```
public ref class OracleDBBinding sealed : public AdapterBinding
```

J#

```
public final class OracleDBBinding extends AdapterBinding
```

JScript

```
public final class OracleDBBinding extends AdapterBinding
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.Binding](#)

[Microsoft.ServiceModel.Channels.Common.AdapterBinding](#)

Microsoft.Adapters.OracleDB.OracleDBBinding

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

OracleDBBinding Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Creates a binding for Oracle DB adapter.

The following tables list the members exposed by the [OracleDBBinding](#) type.

Public Constructors

Name	Description
OracleDBBinding	Overloaded. Initializes a new instance of the OracleDBBinding class.

Top

Public Properties

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 CloseTimeout	(inherited from Binding)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 ConnectionUriType	Overridden. Gets the type of Uri connection.
 DataFetchSize	Gets or sets the amount of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
 EnableBizTalkCompatibilityMode	Gets or sets whether the value of this binding property is enabled when using the adapter with Biztalk Server.
 EnablePerformanceCounters	Gets or sets whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets whether safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the type of inbound operation to be performed.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for record insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size of a long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MessageVersion	(inherited from Binding)
 MetadataPooling	Gets or sets whether metadata pooling is used.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationPort	Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from Binding)
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the transacted polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried. Obsolete.

 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the post poll statement.
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Gets the scheme name for the specified object.
 SendTimeout	(inherited from Binding)
 SkipNilNodes	Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets whether statement cache is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached.
 SupportsCertificateAuthentication	(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Gets a value that indicates whether the specified object supports metadata browse.
 SupportsMetadataGet	Overridden. Gets a value that indicates whether the specified object supports metadata get.
 SupportsMetadataSearch	Overridden. Gets a value that indicates whether the specified object supports metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 TransactionIsolationLevel	Gets or sets the Oracle transaction isolation level. Obsolete.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether oracle connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates a binding element collection and adds an instance of Oracle DB adapter to it.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[OracleDBBinding Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBBinding](#) class.

Overload List

Name	Description
OracleDBBinding ()	Initializes a new instance of the Oracle DB adapter binding class.
OracleDBBinding (String)	Initializes a new instance of the Oracle DB adapter binding class with binding properties specified in the named configuration.

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding Constructor ()

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the Oracle DB adapter binding class.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="Microsoft.Adapters.OracleDB.OracleDBAdapter")> _
Public Sub New
```

C#

```
[SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="Microsoft.Adapters.OracleDB.OracleDBAdapter")]
public OracleDBBinding ()
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Usage", L"CA1806:DoNotIgnoreMethodResults", MessageId=L"Microsoft.Adapters.OracleDB.OracleDBAdapter")]
public:
OracleDBBinding ()
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="Microsoft.Adapters.OracleDB.OracleDBAdapter") */
public OracleDBBinding ()
```

JScript

```
SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="Microsoft.Adapters.OracleDB.OracleDBAdapter")
public function OracleDBBinding ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding Constructor (String)

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the Oracle DB adapter binding class with binding properties specified in the named configuration.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    configurationName As String _  
)
```

C#

```
public OracleDBBinding (  
    string configurationName  
)
```

C++

```
public:  
OracleDBBinding (  
    String^ configurationName  
)
```

J#

```
public OracleDBBinding (  
    String configurationName  
)
```

JScript

```
public function OracleDBBinding (  
    configurationName : String  
)
```

Parameters

configurationName

The configuration name.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates a binding element collection and adds an instance of Oracle DB adapter to it.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleDBBinding Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.CreateBindingElements Method

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Creates a binding element collection and adds an instance of Oracle DB adapter to it.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Overrides NotOverridable Function CreateBindingElements As BindingElementCollection
```

C#

```
public override sealed BindingElementCollection CreateBindingElements ()
```

C++

```
public:  
virtual BindingElementCollection^ CreateBindingElements () override sealed
```

J#

```
public final BindingElementCollection CreateBindingElements ()
```

JScript

```
public override final function CreateBindingElements () : BindingElementCollection
```

Return Value

A collection of binding elements.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

OracleDBBinding Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 CloseTimeout	(inherited from Binding)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 ConnectionUriType	Overridden. Gets the type of Uri connection.
 DataFetchSize	Gets or sets the amount of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
 EnableBizTalkCompatibilityMode	Gets or sets whether the value of this binding property is enabled when using the adapter with Biztalk Server.
 EnablePerformanceCounters	Gets or sets whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets whether safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the type of inbound operation to be performed.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for record insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size of a long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MessageVersion	(inherited from Binding)
 MetadataPooling	Gets or sets whether metadata pooling is used.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationPort	Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from Binding)
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the transacted polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried. Obsolete.
 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the post poll statement.
 QueuedDelivery	(inherited from AdapterBinding)

 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Gets the scheme name for the specified object.
 SendTimeout	(inherited from Binding)
 SkipNilNodes	Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets whether statement cache is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached.
 SupportsCertificateAuthentication	(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Gets a value that indicates whether the specified object supports metadata browse.
 SupportsMetadataGet	Overridden. Gets a value that indicates whether the specified object supports metadata get.
 SupportsMetadataSearch	Overridden. Gets a value that indicates whether the specified object supports metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 TransactionIsolationLevel	Gets or sets the Oracle transaction isolation level. Obsolete.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether oracle connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNamespace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

See Also

Reference

[OracleDBBinding Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.AcceptCredentialsInUri Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the Oracle connection URI can contain user credentials for the Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Security")> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Security")]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Security")]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AcceptCredentialsInUri () : boolean

CompilerGeneratedAttribute
public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle connection URI can contain user credentials for the Oracle

database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.ConnectionLifetime Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("OracleConnectionPool")> _
<BrowsableAttribute(True)> _
Public Property ConnectionLifetime As Integer
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[BrowsableAttribute(true)]
public int ConnectionLifetime { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[BrowsableAttribute(true)]
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int

public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.ConnectionUriType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets the type of Uri connection.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property ConnectionUriType As Type
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override Type ConnectionUriType { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property Type^ ConnectionUriType {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_ConnectionUriType ()
```

JScript

```
public override function get ConnectionUriType () : Type
```

Property Value

The type of Uri connection.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.DataFetchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the amount of data to fetch.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("BufferManagement")> _
Public Property DataFetchSize As Long
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("BufferManagement")]
public long DataFetchSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The data type.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.DecrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property DecrPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int DecrPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.EnableBizTalkCompatibilityMode Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the value of this binding property is enabled when using the adapter with Biztalk Server.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[BrowsableAttribute(true)]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[BrowsableAttribute(true)]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnableBizTalkCompatibilityMode () : boolean

CompilerGeneratedAttribute
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the value of this binding property is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.EnablePerformanceCounters Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the performance counters are enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Diagnostics")> _
<BrowsableAttribute(True)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[BrowsableAttribute(true)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[BrowsableAttribute(true)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnablePerformanceCounters ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnablePerformanceCounters () : boolean

CompilerGeneratedAttribute
public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if performance counters are enabled; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.EnableSafeTyping Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether safe typing is enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Metadada")> _
Public Property EnableSafeTyping As Boolean
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Metadada")]
public bool EnableSafeTyping { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Metadada")]
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnableSafeTyping ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnableSafeTyping () : boolean

CompilerGeneratedAttribute
public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if safe typing is enabled; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.GeneratedUserTypesAssemblyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get GeneratedUserTypesAssemblyFilePath () : String

CompilerGeneratedAttribute
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.GeneratedUserTypesAssemblyKeyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(False, False)> _
<BrowsableAttribute(True)> _
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
[BindingPropertyDirectionAttribute(false, false)]
[BrowsableAttribute(true)]
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, false)]
[BrowsableAttribute(true)]
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
public:
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String

CompilerGeneratedAttribute
public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.InboundOperationType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the type of inbound operation to be performed.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The type of inbound operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.IncrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleConnectionPool")> _
<BrowsableAttribute(True)> _
Public Property IncrPoolSize As Integer
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[BrowsableAttribute(true)]
public int IncrPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[BrowsableAttribute(true)]
public:
virtual property int IncrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_IncrPoolSize ()

/** @property */
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int

public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections to be created.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.InsertBatchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the batch size for record insert operations.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property InsertBatchSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public int InsertBatchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int

public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for record insert operations.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.LongDatatypeColumnSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the maximum size of a long data type column.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")> _
Public Property LongDatatypeColumnSize As Long
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")]
public long LongDatatypeColumnSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation
for suggested alternatives")]
public:
virtual property long long LongDatatypeColumnSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (long value)
```

JScript

```
public final function get LongDatatypeColumnSize () : long
public final function set LongDatatypeColumnSize (value : long)
```

Property Value

The maximum size of a long data type column.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.MaxOutputAssociativeArrayElements Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("BufferManagement")> _
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("BufferManagement")]
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property int MaxOutputAssociativeArrayElements {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxOutputAssociativeArrayElements ()

/** @property */
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get MaxOutputAssociativeArrayElements () : int

public final function set MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array that the binding creates.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.MaxPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MaxPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int MaxPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.MetadataPooling Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether metadata pooling is used.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property MetadataPooling As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public bool MetadataPooling { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_MetadataPooling ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MetadataPooling (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MetadataPooling () : boolean

CompilerGeneratedAttribute
public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if metadata pooling is used; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.MinPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the minimum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MinPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int MinPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MinPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MinPoolSize ()

/** @property */
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int

public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.NotificationPort Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the port number that the data provider must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
<BrowsableAttribute(True)> _
Public Property NotificationPort As Integer
```

C#

```
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
public int NotificationPort { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
public:
virtual property int NotificationPort {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_NotificationPort ()

/** @property */
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int

public final function set NotificationPort (value : int)
```

Property Value

The notification port number.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.NotificationStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<BrowsableAttribute(True)> _
<CategoryAttribute("Notification")> _
Public Property NotificationStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
[CategoryAttribute("Notification")]
public string NotificationStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
[CategoryAttribute(L"Notification")]
public:
virtual property String^ NotificationStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

The notification statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.NotifyOnListenerStart Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[CategoryAttribute("Notification")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the adapter sends a notification message when the listener starts; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PolledDataAvailableStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The polled data available statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PollingAction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PollingAction As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public string PollingAction { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingAction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingAction (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingAction () : String

CompilerGeneratedAttribute
public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PollingInterval Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the transacted polling interval.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PollingInterval As Integer
```

C#

```
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public int PollingInterval { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int

public final function set PollingInterval (value : int)
```

Property Value

The polling interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PollingRetryCount Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the number of times that polling is retried.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
<BindingPropertyDirectionAttribute(False, False)> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
Public Property PollingRetryCount As Integer
```

C#

```
[BrowsableAttribute(false)]
[BindingPropertyDirectionAttribute(false, false)]
[ObsoleteAttribute("This field/property/method is deprecated")]
public int PollingRetryCount { get; set; }
```

C++

```
[BrowsableAttribute(false)]
[BindingPropertyDirectionAttribute(false, false)]
[ObsoleteAttribute("This field/property/method is deprecated")]
public:
property int PollingRetryCount {
    int get ();
    void set (int value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public int get_PollingRetryCount ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_PollingRetryCount (int value)
```

JScript

```
CompilerGeneratedAttribute
public function get PollingRetryCount () : int

CompilerGeneratedAttribute
public function set PollingRetryCount (value : int)
```

Property Value

The number of times that polling is retried.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PollingStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the polling statement.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
Public Property PollingStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
public string PollingStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
public:
virtual property String^ PollingStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingStatement () : String

CompilerGeneratedAttribute
public final function set PollingStatement (value : String)
```

Property Value

The polling statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PollWhileDataFound Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the specified object continuously poll data when it is available.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object continuously poll data when it is available; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.PostPollStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the post poll statement.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PostPollStatement As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public string PostPollStatement { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PostPollStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PostPollStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PostPollStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PostPollStatement () : String

CompilerGeneratedAttribute
public final function set PostPollStatement (value : String)
```

Property Value

The post poll statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.Scheme Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets the scheme name for the specified object.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property Scheme As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override string Scheme { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property String^ Scheme {
String^ get () override;
}
```

J#

```
/** @property */
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Property Value

The scheme name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.SkipNilNodes Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the specified object skips inserting or updating values for nodes that are marked as 'nil' in the request XML.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Run Time Behavior")> _
<BrowsableAttribute(True)> _
Public Property SkipNilNodes As Boolean
```

C#

```
[CategoryAttribute("Run Time Behavior")]
[BrowsableAttribute(true)]
public bool SkipNilNodes { get; set; }
```

C++

```
[CategoryAttribute(L"Run Time Behavior")]
[BrowsableAttribute(true)]
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_SkipNilNodes ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get SkipNilNodes () : boolean

CompilerGeneratedAttribute
public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if inserting nil nodes is skipped; otherwise,
<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.StatementCachePurge Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether statement cache is purged.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("BufferManagement")> _
Public Property StatementCachePurge As Boolean
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("BufferManagement")]
public bool StatementCachePurge { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_StatementCachePurge ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get StatementCachePurge () : boolean

CompilerGeneratedAttribute
public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if statement cache is purged; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.StatementCacheSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum number of statements that can be cached.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property StatementCacheSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public int StatementCacheSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int

public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.SupportsMetadataBrowse Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets a value that indicates whether the specified object supports metadata browse.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property SupportsMetadataBrowse As Boolean
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override bool SupportsMetadataBrowse { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property bool SupportsMetadataBrowse {
    bool get () override;
}
```

J#

```
/** @property */
public boolean get_SupportsMetadataBrowse ()
```

JScript

```
public override function get SupportsMetadataBrowse () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object supports metadata browse; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.SupportsMetadataGet Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets a value that indicates whether the specified object supports metadata get.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property SupportsMetadataGet As Boolean
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override bool SupportsMetadataGet { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property bool SupportsMetadataGet {
    bool get () override;
}
```

J#

```
/** @property */
public boolean get_SupportsMetadataGet ()
```

JScript

```
public override function get SupportsMetadataGet () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object supports metadata get; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.SupportsMetadataSearch Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets a value that indicates whether the specified object supports metadata search.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property SupportsMetadataSearch As Boolean
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override bool SupportsMetadataSearch { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property bool SupportsMetadataSearch {
    bool get () override;
}
```

J#

```
/** @property */
public boolean get_SupportsMetadataSearch ()
```

JScript

```
public override function get SupportsMetadataSearch () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object supports metadata search; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.TransactionIsolationLevel Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the Oracle transaction isolation level.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property TransactionIsolationLevel As TransactionIsolationLevel
```

C#

```
[BrowsableAttribute(false)]
[ObsoleteAttribute("This field/property/method is deprecated")]
[BindingPropertyDirectionAttribute(false, false)]
public TransactionIsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[BrowsableAttribute(false)]
[ObsoleteAttribute(L"This field/property/method is deprecated")]
[BindingPropertyDirectionAttribute(false, false)]
public:
    property TransactionIsolationLevel TransactionIsolationLevel {
        TransactionIsolationLevel get ();
        void set (TransactionIsolationLevel value);
    }
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public TransactionIsolationLevel get_TransactionIsolationLevel ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_TransactionIsolationLevel (TransactionIsolationLevel value)
```

JScript

```
CompilerGeneratedAttribute
public function get TransactionIsolationLevel () : TransactionIsolationLevel

CompilerGeneratedAttribute
public function set TransactionIsolationLevel (value : TransactionIsolationLevel)
```

Property Value

The Oracle transaction isolation level.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.UseAmbientTransaction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Transactions")> _
<BrowsableAttribute(True)> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[CategoryAttribute("Transactions")]
[BrowsableAttribute(true)]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[CategoryAttribute(L"Transactions")]
[BrowsableAttribute(true)]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseAmbientTransaction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseAmbientTransaction () : boolean

CompilerGeneratedAttribute
public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter performs the operations using the transaction

context provided by the caller; otherwise, <languageKeyword>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.UseOracleConnectionPool Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether oracle connection pool is used.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleConnectionPool")> _
<BrowsableAttribute(True)> _
Public Property UseOracleConnectionPool As Boolean
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[BrowsableAttribute(true)]
public bool UseOracleConnectionPool { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[BrowsableAttribute(true)]
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseOracleConnectionPool ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseOracleConnectionPool () : boolean

CompilerGeneratedAttribute
public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if oracle connection pool is used; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.UserAssembliesLoadPath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("UDT .NET Type Generation - Run Time")> _
Public Property UserAssembliesLoadPath As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("UDT .NET Type Generation - Run Time")]
public string UserAssembliesLoadPath { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"UDT .NET Type Generation - Run Time")]
public:
virtual property String^ UserAssembliesLoadPath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_UserAssembliesLoadPath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UserAssembliesLoadPath () : String

CompilerGeneratedAttribute
public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The name of the DLLs.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBinding.UseSchemaInNameSpace Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Metadadata")> _
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Metadadata")]
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Metadadata")]
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseSchemaInNameSpace () : boolean

CompilerGeneratedAttribute
public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the schema name is included in the xml namespace for operations and their

associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBinding Class](#)

[OracleDBBinding Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the configuration element for Oracle DB adapter binding.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
Public NotInheritable Class OracleDBBindingConfigurationElement
    Inherits StandardBindingElement
```

C#

```
public sealed class OracleDBBindingConfigurationElement : StandardBindingElement
```

C++

```
public ref class OracleDBBindingConfigurationElement sealed : public StandardBindingElement
```

J#

```
public final class OracleDBBindingConfigurationElement extends StandardBindingElement
```

JScript

```
public final class OracleDBBindingConfigurationElement extends StandardBindingElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.StandardBindingElement](#)

Microsoft.Adapters.OracleDB.OracleDBBindingConfigurationElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

OracleDBBindingConfigurationElement Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the configuration element for Oracle DB adapter binding.

The following tables list the members exposed by the [OracleDBBindingConfigurationElement](#) type.

Public Constructors

Name	Description
OracleDBBindingConfigurationElement	Overloaded.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets a value whether the Oracle connection URI can contain user credentials for the Oracle database.
 CloseTimeout	(inherited from StandardBindingElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the size of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value that indicates whether the binding property is set to compatibility mode.
 EnablePerformanceCounters	Gets or sets a value whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets a value that indicates whether the safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets whether you want to perform <code><languageKeyword>Polling</languageKeyword></code> or <code><languageKeyword>Notification</languageKeyword></code> inbound operation.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested.
 InsertBatchSize	Gets or sets the batch size to insert.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the column size. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations.
 MaxPoolSize	Gets or sets the maximum number of connections allowed in the connection pool for this specific connection string.
 MetadataPooling	Gets or sets a value whether the MetadataPooling is specified.
 MinPoolSize	Gets or sets the minimum number of connections in a connection pool that the specified connection accepts when establishing a connection with a server.
 Name	(inherited from StandardBindingElement)
 NotificationPort	Gets or sets the port number that the data provider opens to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

 OpenTimeout	(inherited from StandardBindingElement)
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the duration (in seconds) of the database's polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried if an exception is encountered during the polling transaction. Obsolete.
 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.
 PostPollStatement	Gets or sets the statement block that is executed after the statement specified by the PollingInput binding property is executed.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 SkipNilNodes	Gets or sets a value that indicates whether the Oracle Database adapter skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets a value whether the StatementCachePurge is specified.
 StatementCacheSize	Gets or sets the size of the statement cache.
 TransactionIsolationLevel	Gets or sets the transaction isolation level for new transactions created inside the service, and incoming transactions flowed from a client. Obsolete.
 UseAmbientTransaction	Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets a value whether a connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets a value that specifies whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Protected Properties

	Name	Description
	BindingElementType	(inherited from StandardBindingElement)
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingElement)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from StandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)

 MemberwiseClone	(inherited from Object)
 OnApplyConfiguration	(inherited from StandardBindingElement)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from StandardBindingElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Overload List

Name	Description
OracleDBBindingConfigurationElement ()	Initializes a new instance of the OracleDBBindingConfigurationElement class.
OracleDBBindingConfigurationElement (String)	Initializes a new instance of the OracleDBBindingConfigurationElement class with the specified name.

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Constructor ()

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBBindingConfigurationElement](#) class.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleDBBindingConfigurationElement ()
```

C++

```
public:  
OracleDBBindingConfigurationElement ()
```

J#

```
public OracleDBBindingConfigurationElement ()
```

JScript

```
public function OracleDBBindingConfigurationElement ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Constructor (String)

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBBindingConfigurationElement](#) class with the specified name.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    name As String _  
)
```

C#

```
public OracleDBBindingConfigurationElement (  
    string name  
)
```

C++

```
public:  
OracleDBBindingConfigurationElement (  
    String^ name  
)
```

J#

```
public OracleDBBindingConfigurationElement (  
    String name  
)
```

JScript

```
public function OracleDBBindingConfigurationElement (  
    name : String  
)
```

Parameters

name

The name.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from StandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnApplyConfiguration	(inherited from StandardBindingElement)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from StandardBindingElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets a value whether the Oracle connection URI can contain user credentials for the Oracle database.
 CloseTimeout	(inherited from StandardBindingElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the size of data to fetch.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value that indicates whether the binding property is set to compatibility mode.
 EnablePerformanceCounters	Gets or sets a value whether the performance counters are enabled.
 EnableSafeTyping	Gets or sets a value that indicates whether the safe typing is enabled.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets whether you want to perform <code><languageKeyword>Polling</languageKeyword></code> or <code><languageKeyword>Notification</languageKeyword></code> inbound operation.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested.
 InsertBatchSize	Gets or sets the batch size to insert.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the column size. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations.
 MaxPoolSize	Gets or sets the maximum number of connections allowed in the connection pool for this specific connection string.
 MetadataPooling	Gets or sets a value whether the MetadataPooling is specified.
 MinPoolSize	Gets or sets the minimum number of connections in a connection pool that the specified connection accepts when establishing a connection with a server.
 Name	(inherited from StandardBindingElement)
 NotificationPort	Gets or sets the port number that the data provider opens to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from StandardBindingElement)
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInterval	Gets or sets the duration (in seconds) of the database's polling interval.
 PollingRetryCount	Gets or sets the number of times that polling is retried if an exception is encountered during the polling transaction. Obsolete.

 PollingStatement	Gets or sets the polling statement.
 PollWhileDataFound	Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.
 PostPollStatement	Gets or sets the statement block that is executed after the statement specified by the PollingInput binding property is executed.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 SkipNilNodes	Gets or sets a value that indicates whether the Oracle Database adapter skips inserting or updating values for nodes that are marked as 'nil' in the request XML.
 StatementCachePurge	Gets or sets a value whether the StatementCachePurge is specified.
 StatementCacheSize	Gets or sets the size of the statement cache.
 TransactionIsolationLevel	Gets or sets the transaction isolation level for new transactions created inside the service, and incoming transactions flowed from a client. Obsolete.
 UseAmbientTransaction	Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets a value whether a connection pool is used.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets a value that specifies whether the schema name is included in the xml namespace for operations and their associated types.

Protected Properties

	Name	Description
	BindingElementType	(inherited from StandardBindingElement)
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingElement)

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)
[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.AcceptCredentialsInUri Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value whether the Oracle connection URI can contain user credentials for the Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("acceptCredentialsInUri")> _
<BrowsableAttribute(False)> _
<CategoryAttribute("Security")> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[ConfigurationPropertyAttribute("acceptCredentialsInUri")]
[BrowsableAttribute(false)]
[CategoryAttribute("Security")]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"acceptCredentialsInUri")]
[BrowsableAttribute(false)]
[CategoryAttribute(L"Security")]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle connection URI can contain user credentials for the Oracle database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.ConnectionLifetime Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("connectionLifetime")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property ConnectionLifetime As Integer
```

C#

```
[ConfigurationPropertyAttribute("connectionLifetime")]
[CategoryAttribute("OracleConnectionPool")]
public int ConnectionLifetime { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"connectionLifetime")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int
public final function set ConnectionLifetime (value : int)
```

Property Value

An Int32 value that specifies the maximum duration in seconds of a connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.DataFetchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the size of data to fetch.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("dataFetchSize")> _
Public Property DataFetchSize As Long
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("dataFetchSize")]
public long DataFetchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"dataFetchSize")]
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The size of data to fetch.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.DecrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("decrPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property DecrPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("decrPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int DecrPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"decrPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

An Int32 value that specifies the number of connections that are closed when an excessive amount of established connections are not in use.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.EnableBizTalkCompatibilityMode Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the binding property is set to compatibility mode.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode")> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode")]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableBizTalkCompatibilityMode")]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean

public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the binding property is set to compatibility mode; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.EnablePerformanceCounters Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value whether the performance counters are enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Diagnostics")> _
<ConfigurationPropertyAttribute("enablePerformanceCounters")> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[ConfigurationPropertyAttribute("enablePerformanceCounters")]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[ConfigurationPropertyAttribute(L"enablePerformanceCounters")]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if performance counters are enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit

and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.EnableSafeTyping Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the safe typing is enabled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enableSafeTyping")> _
<CategoryAttribute("Metadata")> _
Public Property EnableSafeTyping As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableSafeTyping")]
[CategoryAttribute("Metadata")]
public bool EnableSafeTyping { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableSafeTyping")]
[CategoryAttribute(L"Metadata")]
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableSafeTyping ()

/** @property */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
public final function get EnableSafeTyping () : boolean

public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the safe typing is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.GeneratedUserTypesAssemblyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<BindingPropertyDirectionAttribute(False, False)> _
<ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath")> _
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath")]
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyFilePath")]
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyFilePath () : String
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

ThreadSafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.GeneratedUserTypesAssemblyKeyFilePath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<BindingPropertyDirectionAttribute(False, False)> _
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath")> _
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
[BindingPropertyDirectionAttribute(false, false)]
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath")]
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, false)]
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyKeyFilePath")]
public:
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String

public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.InboundOperationType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets whether you want to perform <languageKeyword>Polling</languageKeyword> or <languageKeyword>Notification</languageKeyword> inbound operation.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("inboundOperationType"> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[ConfigurationPropertyAttribute("inboundOperationType")]
[BindingPropertyDirectionAttribute(true, false)]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"inboundOperationType")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
public final InboundOperation get_InboundOperationType ()

/** @property */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
public final function get InboundOperationType () : InboundOperation
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The inbound operation to perform.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.IncrPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of new connections to be created when a new connection is requested.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("incrPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property IncrPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("incrPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int IncrPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"incrPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int IncrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_IncrPoolSize ()

/** @property */
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int

public final function set IncrPoolSize (value : int)
```

Property Value

An Int32 value that specifies the number of new connections to be created.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.InsertBatchSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the batch size to insert.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("insertBatchSize")> _
<CategoryAttribute("BufferManagement")> _
Public Property InsertBatchSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("insertBatchSize")]
[CategoryAttribute("BufferManagement")]
public int InsertBatchSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"insertBatchSize")]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int

public final function set InsertBatchSize (value : int)
```

Property Value

An Int32 value of the batch size to insert.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.LongDatatypeColumnSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the column size.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("longDatatypeColumnSize")> _
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for
suggested alternatives")> _
Public Property LongDatatypeColumnSize As Long
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("longDatatypeColumnSize")]
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for
suggested alternatives")]
public long LongDatatypeColumnSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"longDatatypeColumnSize")]
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation for
suggested alternatives")]
public:
virtual property long long LongDatatypeColumnSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (long value)
```

JScript

```
public final function get LongDatatypeColumnSize () : long
public final function set LongDatatypeColumnSize (value : long)
```

Property Value

An Int64 value of the column size.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.MaxOutputAssociativeArrayElements Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the size of the associate array that the adapter creates when performing operations.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements")> _
<CategoryAttribute("BufferManagement")> _
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements")]
[CategoryAttribute("BufferManagement")]
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxOutputAssociativeArrayElements")]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property int MaxOutputAssociativeArrayElements {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxOutputAssociativeArrayElements ()

/** @property */
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get MaxOutputAssociativeArrayElements () : int

public final function set MaxOutputAssociativeArrayElements (value : int)
```

Property Value

An Int32 value that specifies the size of the associate array that the adapter creates when performing operations.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

OracleDBBindingConfigurationElement.MaxPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the maximum number of connections allowed in the connection pool for this specific connection string.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("maxPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MaxPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int MaxPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

An Int32 value that specifies the maximum number of connections in a connection pool.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.MetadataPooling Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value whether the MetadataPooling is specified.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("metadataPooling")> _
Public Property MetadataPooling As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("metadataPooling")]
public bool MetadataPooling { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"metadataPooling")]
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_MetadataPooling ()

/** @property */
public final void set_MetadataPooling (boolean value)
```

JScript

```
public final function get MetadataPooling () : boolean

public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the MetadataPooling is specified; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.MinPoolSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the minimum number of connections in a connection pool that the specified connection accepts when establishing a connection with a server.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("minPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MinPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("minPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int MinPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"minPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MinPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MinPoolSize ()

/** @property */
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int

public final function set MinPoolSize (value : int)
```

Property Value

An Int32 value that specifies the minimum number of connections in a connection pool.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.NotificationPort Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the port number that the data provider opens to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("notificationPort")> _
<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotificationPort As Integer
```

C#

```
[ConfigurationPropertyAttribute("notificationPort")]
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public int NotificationPort { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notificationPort")]
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property int NotificationPort {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_NotificationPort ()

/** @property */
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int

public final function set NotificationPort (value : int)
```

Property Value

An Int32 value of the port number that the data provider opens.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.NotificationStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement used to register for getting notifications form Oracle database.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("notificationStatement")> _
<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotificationStatement As String
```

C#

```
[ConfigurationPropertyAttribute("notificationStatement")]
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public string NotificationStatement { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notificationStatement")]
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NotificationStatement ()

/** @property */
public final void set_NotificationStatement (String value)
```

JScript

```
public final function get NotificationStatement () : String
public final function set NotificationStatement (value : String)
```

Property Value

The SELECT statement used to register for getting notifications from Oracle database.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.NotifyOnListenerStart Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("notifyOnListenerStart")> _
<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[ConfigurationPropertyAttribute("notifyOnListenerStart")]
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notifyOnListenerStart")]
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NotifyOnListenerStart ()

/** @property */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
public final function get NotifyOnListenerStart () : boolean
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the adapter sends a notification message to the adapter clients; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PolledDataAvailableStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("PollingReceive")> _
<ConfigurationPropertyAttribute("polledDataAvailableStatement")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[ConfigurationPropertyAttribute("polledDataAvailableStatement")]
[BindingPropertyDirectionAttribute(true, false)]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[ConfigurationPropertyAttribute(L"polledDataAvailableStatement")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PolledDataAvailableStatement ()

/** @property */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
public final function get PolledDataAvailableStatement () : String

public final function set PolledDataAvailableStatement (value : String)
```

Property Value

A value for the statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit

editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PollingAction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<ConfigurationPropertyAttribute("pollingAction")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PollingAction As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[ConfigurationPropertyAttribute("pollingAction")]
[BindingPropertyDirectionAttribute(true, false)]
public string PollingAction { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[ConfigurationPropertyAttribute(L"pollingAction")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingAction ()

/** @property */
public final void set_PollingAction (String value)
```

JScript

```
public final function get PollingAction () : String

public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PollingInterval Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the duration (in seconds) of the database's polling interval.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingInterval")> _
<CategoryAttribute("PollingReceive")> _
Public Property PollingInterval As Integer
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingInterval")]
[CategoryAttribute("PollingReceive")]
public int PollingInterval { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingInterval")]
[CategoryAttribute(L"PollingReceive")]
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int
public final function set PollingInterval (value : int)
```

Property Value

The duration of the database's polling interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PollingRetryCount Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the number of times that polling is retried if an exception is encountered during the polling transaction.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
<BindingPropertyDirectionAttribute(False, False)> _
<SuppressMessageAttribute("Microsoft.Performance", "CA1822:MarkMembersAsStatic")> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
Public Property PollingRetryCount As Integer
```

C#

```
[BrowsableAttribute(false)]
[BindingPropertyDirectionAttribute(false, false)]
[SuppressMessageAttribute("Microsoft.Performance", "CA1822:MarkMembersAsStatic")]
[ObsoleteAttribute("This field/property/method is deprecated")]
public int PollingRetryCount { get; set; }
```

C++

```
[BrowsableAttribute(false)]
[BindingPropertyDirectionAttribute(false, false)]
[SuppressMessageAttribute(L"Microsoft.Performance", L"CA1822:MarkMembersAsStatic")]
[ObsoleteAttribute(L"This field/property/method is deprecated")]
public:
property int PollingRetryCount {
    int get ();
    void set (int value);
}
```

J#

```
/** @property */
public int get_PollingRetryCount ()

/** @property */
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA1801:ReviewUnusedParameters",
MessageId="value") */
public void set_PollingRetryCount (int value)
```

JScript

```
public function get PollingRetryCount () : int
```

```
SuppressMessageAttribute("Microsoft.Usage", "CA1801:ReviewUnusedParameters", MessageId="value")  
public function set PollingRetryCount (value : int)
```

Property Value

The number of times that polling is retried.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PollingStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the polling statement.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("PollingReceive")> _
<ConfigurationPropertyAttribute("pollingStatement")> _
Public Property PollingStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("PollingReceive")]
[ConfigurationPropertyAttribute("pollingStatement")]
public string PollingStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"PollingReceive")]
[ConfigurationPropertyAttribute(L"pollingStatement")]
public:
virtual property String^ PollingStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingStatement ()

/** @property */
public final void set_PollingStatement (String value)
```

JScript

```
public final function get PollingStatement () : String
public final function set PollingStatement (value : String)
```

Property Value

The polling statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PollWhileDataFound Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<ConfigurationPropertyAttribute("pollWhileDataFound")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[CategoryAttribute("PollingReceive")]
[ConfigurationPropertyAttribute("pollWhileDataFound")]
[BindingPropertyDirectionAttribute(true, false)]
public bool PollWhileDataFound { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[ConfigurationPropertyAttribute(L"pollWhileDataFound")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_PollWhileDataFound ()

/** @property */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
public final function get PollWhileDataFound () : boolean

public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter ignores the polling interval; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.PostPollStatement Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the statement block that is executed after the statement specified by the PollingInput binding property is executed.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("PollingReceive")> _
<ConfigurationPropertyAttribute("postPollStatement")> _
Public Property PostPollStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("PollingReceive")]
[ConfigurationPropertyAttribute("postPollStatement")]
public string PostPollStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"PollingReceive")]
[ConfigurationPropertyAttribute(L"postPollStatement")]
public:
virtual property String^ PostPollStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PostPollStatement ()

/** @property */
public final void set_PostPollStatement (String value)
```

JScript

```
public final function get PostPollStatement () : String
public final function set PostPollStatement (value : String)
```

Property Value

The statement block that is executed after the statement specified by the PollingInput binding property is executed.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.SkipNilNodes Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the Oracle Database adapter skips inserting or updating values for nodes that are marked as 'nil' in the request XML.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("skipNilNodes")> _
<CategoryAttribute("Run Time Behavior")> _
Public Property SkipNilNodes As Boolean
```

C#

```
[ConfigurationPropertyAttribute("skipNilNodes")]
[CategoryAttribute("Run Time Behavior")]
public bool SkipNilNodes { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"skipNilNodes")]
[CategoryAttribute(L"Run Time Behavior")]
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_SkipNilNodes ()

/** @property */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
public final function get SkipNilNodes () : boolean

public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter skips inserting or updating values for nodes that are marked as 'nil' in the request XML; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.StatementCachePurge Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value whether the StatementCachePurge is specified.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("statementCachePurge")> _
Public Property StatementCachePurge As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("statementCachePurge")]
public bool StatementCachePurge { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"statementCachePurge")]
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_StatementCachePurge ()

/** @property */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
public final function get StatementCachePurge () : boolean

public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the StatementCachePurge is specified; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.StatementCacheSize Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the size of the statement cache.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("statementCacheSize")> _
Public Property StatementCacheSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("statementCacheSize")]
public int StatementCacheSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"statementCacheSize")]
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int

public final function set StatementCacheSize (value : int)
```

Property Value

A value of the statement cache size.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.TransactionIsolationLevel Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

NOTE: This property is now obsolete.

Gets or sets the transaction isolation level for new transactions created inside the service, and incoming transactions flowed from a client.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<BindingPropertyDirectionAttribute(False, False)> _
<SuppressMessageAttribute("Microsoft.Performance", "CA1822:MarkMembersAsStatic")> _
<ObsoleteAttribute("This field/property/method is deprecated")> _
<BrowsableAttribute(False)> _
Public Property TransactionIsolationLevel As TransactionIsolationLevel
```

C#

```
[BindingPropertyDirectionAttribute(false, false)]
[SuppressMessageAttribute("Microsoft.Performance", "CA1822:MarkMembersAsStatic")]
[ObsoleteAttribute("This field/property/method is deprecated")]
[BrowsableAttribute(false)]
public TransactionIsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, false)]
[SuppressMessageAttribute(L"Microsoft.Performance", L"CA1822:MarkMembersAsStatic")]
[ObsoleteAttribute(L"This field/property/method is deprecated")]
[BrowsableAttribute(false)]
public:
property TransactionIsolationLevel TransactionIsolationLevel {
    TransactionIsolationLevel get ();
    void set (TransactionIsolationLevel value);
}
```

J#

```
/** @property */
public TransactionIsolationLevel get_TransactionIsolationLevel ()

/** @property */
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA1801:ReviewUnusedParameters", MessageId="value") */
public void set_TransactionIsolationLevel (TransactionIsolationLevel value)
```

JScript

```
public function get TransactionIsolationLevel () : TransactionIsolationLevel
```

```
SuppressMessageAttribute("Microsoft.Usage", "CA1801:ReviewUnusedParameters", MessageId="value"
)
public function set TransactionIsolationLevel (value : TransactionIsolationLevel)
```

Property Value

An IsolationLevel value that specifies the transaction isolation level.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.UseAmbientTransaction Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("useAmbientTransaction")> _
<CategoryAttribute("Transactions")> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useAmbientTransaction")]
[CategoryAttribute("Transactions")]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useAmbientTransaction")]
[CategoryAttribute(L"Transactions")]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if Oracle Database adapter performs the operations using the transaction context provided by the caller; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.UseOracleConnectionPool Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value whether a connection pool is used.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("OracleConnectionPool")> _
<ConfigurationPropertyAttribute("useOracleConnectionPool")> _
Public Property UseOracleConnectionPool As Boolean
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[ConfigurationPropertyAttribute("useOracleConnectionPool")]
public bool UseOracleConnectionPool { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[ConfigurationPropertyAttribute(L"useOracleConnectionPool")]
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseOracleConnectionPool ()

/** @property */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
public final function get UseOracleConnectionPool () : boolean

public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if a connection pool is used; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.UserAssembliesLoadPath Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

Declaration

```
<CategoryAttribute("UDT .NET Type Generation - Run Time")> _
<ConfigurationPropertyAttribute("userAssembliesLoadPath")> _
Public Property UserAssembliesLoadPath As String
```

C#

```
[CategoryAttribute("UDT .NET Type Generation - Run Time")]
[ConfigurationPropertyAttribute("userAssembliesLoadPath")]
public string UserAssembliesLoadPath { get; set; }
```

C++

```
[CategoryAttribute(L"UDT .NET Type Generation - Run Time")]
[ConfigurationPropertyAttribute(L"userAssembliesLoadPath")]
public:
virtual property String^ UserAssembliesLoadPath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_UserAssembliesLoadPath ()

/** @property */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
public final function get UserAssembliesLoadPath () : String
public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The name of the DLLs.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBBindingConfigurationElement.UseSchemaInNameSpace Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets a value that specifies whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

Declaration

```
<CategoryAttribute("Metadata")> _
<ConfigurationPropertyAttribute("useSchemaInNameSpace")> _
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
[CategoryAttribute("Metadata")]
[ConfigurationPropertyAttribute("useSchemaInNameSpace")]
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
[CategoryAttribute(L"Metadata")]
[ConfigurationPropertyAttribute(L"useSchemaInNameSpace")]
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
public final function get UseSchemaInNameSpace () : boolean

public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if schema name is included in the xml namespace for operations and their associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBBindingConfigurationElement Class](#)

[OracleDBBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the class for building the OracleDBConnectionUri.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class OracleDBConnectionUri  
    Inherits ConnectionUri
```

C#

```
public sealed class OracleDBConnectionUri : ConnectionUri
```

C++

```
public ref class OracleDBConnectionUri sealed : public ConnectionUri
```

J#

```
public final class OracleDBConnectionUri extends ConnectionUri
```

JScript

```
public final class OracleDBConnectionUri extends ConnectionUri
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.ServiceModel.Channels.Common.ConnectionUri](#)

Microsoft.Adapters.OracleDB.OracleDBConnectionUri

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Members](#)

OracleDBConnectionUri Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the class for building the OracleDBConnectionUri.

The following tables list the members exposed by the [OracleDBConnectionUri](#) type.

Public Constructors

Name	Description
OracleDBConnectionUri	Initializes a new instance of the OracleDBConnectionUri class.

Top

Public Properties

Name	Description
 DataSourceName	Gets or sets the name of the data source.
 PollingId	Gets or sets the specified value of the polling id.
 PortNumber	Gets or sets the number of the port used to communicate with the server.
 SampleUriString	Overridden. Gets a sample URI string.
 SecureUriString	(inherited from ConnectionUri)
 ServerAddress	Gets or sets the address of the server to which connectivity is to be verified.
 ServiceName	Gets or sets the short name used to identify the service to the system.
 ServiceType	Gets or sets the type of the service.
 Uri	Overridden. Gets or sets the <languageKeyword>uniform resource identifier</languageKeyword> used in the connection.

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetAttributes	(inherited from ConnectionUri)
 GetClassName	(inherited from ConnectionUri)
 GetComponentName	(inherited from ConnectionUri)
 GetConverter	(inherited from ConnectionUri)
 GetDefaultEvent	(inherited from ConnectionUri)
 GetDefaultProperty	(inherited from ConnectionUri)
 GetEditor	(inherited from ConnectionUri)
 GetEvents	Overloaded. (inherited from ConnectionUri)
 GetHashCode	(inherited from Object)
 GetProperties	Overloaded. (inherited from ConnectionUri)
 GetPropertyOwner	(inherited from ConnectionUri)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from ConnectionUri)

Top

Protected Methods

Name	Description
 Finalize	(inherited from Object)
  GetQueryStringValue	(inherited from ConnectionUri)
  GetUserInfoValue	(inherited from ConnectionUri)
 MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[OracleDBConnectionUri Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Initializes a new instance of the [OracleDBConnectionUri](#) class.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")>
_
Public Sub New
```

C#

```
[SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")]
public OracleDBConnectionUri ()
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Performance", L"CA1805:DoNotInitializeUnnecessarily")
]
public:
OracleDBConnectionUri ()
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily") */
public OracleDBConnectionUri ()
```

JScript

```
SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")
public function OracleDBConnectionUri ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetAttributes	(inherited from ConnectionUri)
	GetClassName	(inherited from ConnectionUri)
	GetComponentName	(inherited from ConnectionUri)
	GetConverter	(inherited from ConnectionUri)
	GetDefaultEvent	(inherited from ConnectionUri)
	GetDefaultProperty	(inherited from ConnectionUri)
	GetEditor	(inherited from ConnectionUri)
	GetEvents	Overloaded. (inherited from ConnectionUri)
	GetHashCode	(inherited from Object)
	GetProperties	Overloaded. (inherited from ConnectionUri)
	GetPropertyOwner	(inherited from ConnectionUri)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from ConnectionUri)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
 	GetQueryStringValue	(inherited from ConnectionUri)
 	GetUserInfoValue	(inherited from ConnectionUri)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleDBConnectionUri Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties

Name	Description
 DataSourceName	Gets or sets the name of the data source.
 PollingId	Gets or sets the specified value of the polling id.
 PortNumber	Gets or sets the number of the port used to communicate with the server.
 SampleUriString	Overridden. Gets a sample URI string.
 SecureUriString	(inherited from ConnectionUri)
 ServerAddress	Gets or sets the address of the server to which connectivity is to be verified.
 ServiceName	Gets or sets the short name used to identify the service to the system.
 ServiceType	Gets or sets the type of the service.
 Uri	Overridden. Gets or sets the <languageKeyword>uniform resource identifier </languageKeyword> used in the connection.

See Also

Reference

[OracleDBConnectionUri Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.DataSourceName Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the name of the data source.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Oracle Server")> _
<BrowsableAttribute(True)> _
<DescriptionAttribute("Oracle TNS Alias Name")> _
Public Property DataSourceName As String
```

C#

```
[CategoryAttribute("Oracle Server")]
[BrowsableAttribute(true)]
[DescriptionAttribute("Oracle TNS Alias Name")]
public string DataSourceName { get; set; }
```

C++

```
[CategoryAttribute(L"Oracle Server")]
[BrowsableAttribute(true)]
[DescriptionAttribute(L"Oracle TNS Alias Name")]
public:
property String^ DataSourceName {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_DataSourceName ()

/** @property */
public void set_DataSourceName (String value)
```

JScript

```
public function get DataSourceName () : String

public function set DataSourceName (value : String)
```

Property Value

The name of the data source.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.PollingId Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the specified value of the polling id.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("PollingReceive")> _
<DescriptionAttribute("Polling Id")> _
<BrowsableAttribute(True)> _
Public Property PollingId As String
```

C#

```
[CategoryAttribute("PollingReceive")]
[DescriptionAttribute("Polling Id")]
[BrowsableAttribute(true)]
public string PollingId { get; set; }
```

C++

```
[CategoryAttribute(L"PollingReceive")]
[DescriptionAttribute(L"Polling Id")]
[BrowsableAttribute(true)]
public:
property String^ PollingId {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_PollingId ()

/** @property */
public void set_PollingId (String value)
```

JScript

```
public function get PollingId () : String

public function set PollingId (value : String)
```

Property Value

A string value of the polling id.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.PortNumber Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the number of the port used to communicate with the server.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<DefaultValueAttribute(1521)> _
<BrowsableAttribute(True)> _
<CategoryAttribute("Connect Descriptor")> _
Public Property PortNumber As Integer
```

C#

```
[DefaultValueAttribute(1521)]
[BrowsableAttribute(true)]
[CategoryAttribute("Connect Descriptor")]
public int PortNumber { get; set; }
```

C++

```
[DefaultValueAttribute(1521)]
[BrowsableAttribute(true)]
[CategoryAttribute(L"Connect Descriptor")]
public:
property int PortNumber {
    int get ();
    void set (int value);
}
```

J#

```
/** @property */
public int get_PortNumber ()

/** @property */
public void set_PortNumber (int value)
```

JScript

```
public function get PortNumber () : int

public function set PortNumber (value : int)
```

Property Value

An [Int32](#) value that contains the port number used to communicate with the server.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.SampleUriString Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets a sample URI string.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<SuppressMessageAttribute("Microsoft.Design", "CA1056:UriPropertiesShouldNotBeStrings")> _
<SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")> _
Public Overrides ReadOnly Property SampleUriString As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1056:UriPropertiesShouldNotBeStrings")]
[SuppressMessageAttribute("Microsoft.Design", "CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public override string SampleUriString { get; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1056:UriPropertiesShouldNotBeStrings")]
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1048:DoNotDeclareVirtualMembersInSealedTypes")]
public:
virtual property String^ SampleUriString {
String^ get () override;
}
```

J#

```
/** @property */
public String get_SampleUriString ()
```

JScript

```
public override function get SampleUriString () : String
```

Property Value

Returns a String containing a sample URI string.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.ServerAddress Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the address of the server to which connectivity is to be verified.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServerAddress As String
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public string ServerAddress { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property String^ ServerAddress {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public String get_ServerAddress ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServerAddress (String value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServerAddress () : String

CompilerGeneratedAttribute
public function set ServerAddress (value : String)
```

Property Value

The address of the server.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.ServiceName Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the short name used to identify the service to the system.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServiceName As String
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public string ServiceName { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property String^ ServiceName {
    String^ get ();
    void set (String^ value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public String get_ServiceName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServiceName (String value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServiceName () : String

CompilerGeneratedAttribute
public function set ServiceName (value : String)
```

Property Value

The name of the service.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.ServiceType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the type of the service.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServiceType As OracleServiceType
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public OracleServiceType ServiceType { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property OracleServiceType ServiceType {
    OracleServiceType get ();
    void set (OracleServiceType value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public OracleServiceType get_ServiceType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServiceType (OracleServiceType value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServiceType () : OracleServiceType

CompilerGeneratedAttribute
public function set ServiceType (value : OracleServiceType)
```

Property Value

The type of the service.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBConnectionUri.Uri Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the <languageKeyword>uniform resource identifier</languageKeyword> used in the connection.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property Uri As Uri
```

C#

```
public override Uri Uri { get; set; }
```

C++

```
public:  
virtual property Uri^ Uri {  
    Uri^ get () override;  
    void set (Uri^ value) override;  
}
```

J#

```
/** @property */  
public Uri get Uri ()  
  
/** @property */  
public void set Uri (Uri value)
```

JScript

```
public override function get Uri () : Uri  
  
public override function set Uri (value : Uri)
```

Property Value

The URI used in the connection.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBConnectionUri Class](#)

[OracleDBConnectionUri Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the behavior that controls the transaction settings for WCF-Dispatcher initiated transactions.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
Public NotInheritable Class OracleDBInboundTransactionBehavior
    Inherits BehaviorExtensionElement
    Implements IServiceBehavior
```

C#

```
public sealed class OracleDBInboundTransactionBehavior : BehaviorExtensionElement, IServiceBehavior
```

C++

```
public ref class OracleDBInboundTransactionBehavior sealed : public BehaviorExtensionElement, IServiceBehavior
```

J#

```
public final class OracleDBInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

JScript

```
public final class OracleDBInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BehaviorExtensionElement](#)

Microsoft.Adapters.OracleDB.OracleDBInboundTransactionBehavior

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Represents the behavior that controls the transaction settings for WCF-Dispatcher initiated transactions.

The following tables list the members exposed by the [OracleDBInboundTransactionBehavior](#) type.

Public Constructors

Name	Description
OracleDBInboundTransactionBehavior	Default Constructor.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BehaviorType	Overridden. Gets the type of behavior.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 TransactionIsolationLevel	Gets or sets the default isolation level for transactions.
 TransactionTimeout	Gets or sets the period within which a transaction must complete.

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from ConfigurationElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 CopyFrom	Overridden. Copies the content of the specified configuration element to this configuration element.
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 CreateBehavior	(inherited from BehaviorExtensionElement)
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ServiceModelExtensionElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)

 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ServiceModelExtensionElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ServiceModelExtensionElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

[Top](#)

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior Constructor

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Default Constructor.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleDBInboundTransactionBehavior ()
```

C++

```
public:  
OracleDBInboundTransactionBehavior ()
```

J#

```
public OracleDBInboundTransactionBehavior ()
```

JScript

```
public function OracleDBInboundTransactionBehavior ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	CopyFrom	Overridden. Copies the content of the specified configuration element to this configuration element.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBehavior	(inherited from BehaviorExtensionElement)
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.CopyFrom Method

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Copies the content of the specified configuration element to this configuration element.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

The configuration element to be copied.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.AddBinding Method

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JSript

Parameters

serviceDescription

serviceHostBase

endpoints

bindingParameters

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior Method

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JSript

Parameters

serviceDescription

serviceHostBase

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.Validate Method

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior Properties

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Properties (see also [Protected Properties](#))

	Name	Description
	BehaviorType	Overridden. Gets the type of behavior.
	ConfigurationElementName	(inherited from ServiceModelExtensionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	TransactionIsolationLevel	Gets or sets the default isolation level for transactions.
	TransactionTimeout	Gets or sets the period within which a transaction must complete.

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from ConfigurationElement)

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.BehaviorType Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets the type of behavior.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property BehaviorType As Type
```

C#

```
public override Type BehaviorType { get; }
```

C++

```
public:
virtual property Type^ BehaviorType {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_BehaviorType ()
```

JScript

```
public override function get BehaviorType () : Type
```

Property Value

A [Type](#).

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.TransactionIsolationLevel Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the default isolation level for transactions.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue:=IsolationLevel.Serializable)> _
Public Property TransactionIsolationLevel As IsolationLevel
```

C#

```
[ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue=IsolationLevel.Serializable)]
public IsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionIsolationLevel", DefaultValue=IsolationLevel::Serializable)]
public:
property IsolationLevel TransactionIsolationLevel {
    IsolationLevel get ();
    void set (IsolationLevel value);
}
```

J#

```
/** @property */
public IsolationLevel get_TransactionIsolationLevel ()

/** @property */
public void set_TransactionIsolationLevel (IsolationLevel value)
```

JScript

```
public function get TransactionIsolationLevel () : IsolationLevel
public function set TransactionIsolationLevel (value : IsolationLevel)
```

Property Value

The default isolation level for transactions.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBInboundTransactionBehavior.TransactionTimeout Property

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Gets or sets the period within which a transaction must complete.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("transactionTimeout")> _
Public Property TransactionTimeout As TimeSpan
```

C#

```
[ConfigurationPropertyAttribute("transactionTimeout")]
public TimeSpan TransactionTimeout { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionTimeout")]
public:
property TimeSpan TransactionTimeout {
    TimeSpan get ();
    void set (TimeSpan value);
}
```

J#

```
/** @property */
public TimeSpan get_TransactionTimeout ()

/** @property */
public void set_TransactionTimeout (TimeSpan value)
```

JScript

```
public function get TransactionTimeout () : TimeSpan

public function set TransactionTimeout (value : TimeSpan)
```

Property Value

The period within which a transaction must complete.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBInboundTransactionBehavior Class](#)

[OracleDBInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants Class

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Specifies a class exposing constants pertaining to OracleDB Adapter.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public NotInheritable Class OracleDBPerformanceCounterConstants
```

C#

```
public static class OracleDBPerformanceCounterConstants
```

C++

```
public ref class OracleDBPerformanceCounterConstants abstract sealed
```

J#

```
public final class OracleDBPerformanceCounterConstants
```

JScript

```
public final class OracleDBPerformanceCounterConstants
```

Inheritance Hierarchy [System.Object](#)

Microsoft.Adapters.OracleDB.OracleDBPerformanceCounterConstants

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBPerformanceCounterConstants Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants Members

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Specifies a class exposing constants pertaining to OracleDB Adapter.

The following tables list the members exposed by the [OracleDBPerformanceCounterConstants](#) type.

Public Fields

	Name	Description
 	PerformanceCategory	A performance counter category for OracleDB Adapter specific counters.
 	PerformanceCounterLobLatency	The name of the performance counter that tracks the total time spent in the LOB (or LOB client APIs) for a specific action.

Top

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Top

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[OracleDBPerformanceCounterConstants Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants Fields

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Fields

	Name	Description
	PerformanceCategory	A performance counter category for OracleDB Adapter specific counters.
	PerformanceCounterLobLatency	The name of the performance counter that tracks the total time spent in the LOB (or LOB client APIs) for a specific action.

See Also

Reference

[OracleDBPerformanceCounterConstants Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants.PerformanceCategory Field

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

A performance counter category for OracleDB Adapter specific counters.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Const PerformanceCategory As String
```

C#

```
public const string PerformanceCategory
```

C++

```
public:  
literal String^ PerformanceCategory
```

J#

```
public static final String PerformanceCategory
```

JScript

```
public const var PerformanceCategory : String
```

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBPerformanceCounterConstants Class](#)

[OracleDBPerformanceCounterConstants Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants.PerformanceCounterLobLatency Field

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

The name of the performance counter that tracks the total time spent in the LOB (or LOB client APIs) for a specific action.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration  
Public Const PerformanceCounterLobLatency As String
```

C#

```
public const string PerformanceCounterLobLatency
```

C++

```
public:  
literal String^ PerformanceCounterLobLatency
```

J#

```
public static final String PerformanceCounterLobLatency
```

JScript

```
public const var PerformanceCounterLobLatency : String
```

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleDBPerformanceCounterConstants Class](#)

[OracleDBPerformanceCounterConstants Members](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleDBPerformanceCounterConstants Methods

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleDBPerformanceCounterConstants Class](#)

[Microsoft.Adapters.OracleDB Namespace](#)

OracleServiceType Enumeration

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Specifies the type of Oracle database service.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Enumeration OracleServiceType
```

C#

```
public enum OracleServiceType
```

C++

```
public enum class OracleServiceType
```

J#

```
public enum OracleServiceType
```

JScript

```
public enum OracleServiceType
```

Members

Member name	Description
Dedicated	A Dedicated Oracle service is to be used.
Shared	A Shared Oracle service is to be used.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleDB Namespace](#)

TransactionIsolationLevel Enumeration

[This is prerelease documentation and is subject to change in future releases. Blank topics are included as placeholders.]

Specifies the isolation level of the Transaction.

Namespace: Microsoft.Adapters.OracleDB

Assembly: Microsoft.Adapters.OracleDB (in microsoft.adapters.oracledb.dll)

Syntax

VB

```
'Declaration
Public Enumeration TransactionIsolationLevel
```

C#

```
public enum TransactionIsolationLevel
```

C++

```
public enum class TransactionIsolationLevel
```

J#

```
public enum TransactionIsolationLevel
```

JScript

```
public enum TransactionIsolationLevel
```

Members

Member name	Description
ReadCommitted	Defines shared locks that are held while the data is being read to avoid reading modified data.
Serializable	Prevents updating or inserting until the transaction is complete.

Remarks

This enum is not used, since the binding property TransactionIsolationLevel is deprecated.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleDB Namespace](#)

Microsoft BizTalk Adapter for Oracle E-Business Suite Help

Welcome to Microsoft BizTalk Adapter for Oracle E-Business Suite, which includes a variety of resources that can help you learn to develop, deploy, administer, and use the Oracle E-Business adapter.

To download a copy of this Help, go to <http://go.microsoft.com/fwlink/?LinkId=196849>.

	Getting Started Information about new features, limitations, prerequisites, and topics for users who are new to the adapter.
	Oracle E-Business Adapter Tutorial Step-by-step instructions for learning how to use the adapter for specific scenarios.
	Planning and Architecture Architectural overview of the adapter.
	Development Instructions on how to use the adapter with BizTalk Server, the Windows Communication Foundation (WCF) service model, and the WCF channel model.
	Deployment Information about installing the adapter, migrating from previous versions of the adapter, and troubleshooting the installation.
	Troubleshooting Information about enabling tracing and about troubleshooting issues with the BizTalk Adapter Pack and with the adapter, in particular.
	Samples Samples demonstrating how to use the adapter.
	Community Resources Information about how to get assistance with the BizTalk Adapter Pack.

Getting Started

This section provides an overview of the adapter, prerequisites, and topics for users who are new to Microsoft BizTalk Adapter Pack. Information is provided about the features of Microsoft BizTalk Adapter for Oracle E-Business Suite and the different operations that can be performed on the Oracle database by using the adapter.

What is an adapter?

An adapter is a software component that enables you to send and receive messages to and from a line-of-business (LOB) system. The primary goal of an adapter is to facilitate the exchange of business documents between trading partners. Because each business system may adhere to specific document formats and protocols, the adapter must use a delivery mechanism that conforms to commonly recognized standards and protocols to provide a uniform interface to the users.

The adapters in the BizTalk Adapter Pack can be divided into two broad categories:

- **LOB adapters.** LOB adapters provide a service-oriented programming model to access LOB systems—for example, adapters for SAP or Siebel applications.
- **Data adapters.** Data adapters provide a service-oriented programming model to access databases—for example, adapters for the Oracle database or SQL Server.

There are five adapters in the BizTalk Adapter Pack:

- Microsoft BizTalk Adapter for Oracle E-Business Suite (Oracle E-Business adapter)
- Microsoft BizTalk Adapter for SQL Server (SQL adapter)
- Microsoft BizTalk Adapter for Oracle Database (Oracle Database adapter)
- Microsoft BizTalk Adapter for mySAP Business Suite (SAP adapter)
- Microsoft BizTalk Adapter for Siebel eBusiness Applications (Siebel adapter)

 Note
The Siebel adapter is not available for 64-bit platforms.

If you do not already know how you want to use the Oracle E-Business adapter at your company, we recommend that you start by exploring the features and functionality of the adapter described in [Understanding BizTalk Adapter for Oracle E-Business Suite](#).

In This Section

- [Prerequisite Skills and Knowledge](#)
- [Common Administrative Tasks](#)
- [Common Developer Tasks](#)
- [Understanding BizTalk Adapter for Oracle E-Business Suite](#)
- [Oracle E-Business Adapter Tutorial](#)
- [Community Resources](#)
- [Frequently Asked Questions](#)

Prerequisite Skills and Knowledge

The potential users for the BizTalk Adapter Pack are:

- Developers who directly program to the adapters.
- IT professionals who consume the adapters using other integration platforms like Microsoft BizTalk Server and other tools.
- Independent software vendors (ISVs) who build solutions on top of the adapters.

The prerequisite skills and knowledge required for each role are detailed below.

Prerequisites for Developers

A developer using Microsoft BizTalk Adapter for Oracle E-Business Suite must be at least moderately experienced with the use of:

- Microsoft Visual Studio 2010 , with the development of .NET solutions
- Programming with the .NET Framework 4 or .NET Framework 3.5
- Programming with the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
- Extensible Markup Language (XML)
- XML Schema Definition (XSD) language
- Web Services Definition Language (WSDL)

Prerequisites for IT Professionals

An IT professional using the Microsoft BizTalk Adapter for Oracle E-Business Suite must be at least moderately experienced with the use of:

- SQL Server Integration Services (SSIS)
- Microsoft BizTalk Server
- Windows SharePoint Services

Prerequisites for ISVs

ISVs using the Microsoft BizTalk Adapter for Oracle E-Business Suite must be at least moderately experienced with the use of:

- The internal workings and concepts of Oracle E-Business Suite to be able to build applications on top of the adapters
- .NET Framework 4 or .NET Framework 3.5
- The WCF LOB Adapter SDK

See Also

Other Resources

[Getting Started](#)

[Understanding BizTalk Adapter for Oracle E-Business Suite](#)

Common Administrative Tasks

The following table provides links to topics helpful for deploying, using, and maintaining Microsoft BizTalk Adapter for Oracle E-Business Suite.

Administrative Tasks <ul style="list-style-type: none">• Oracle E-Business Adapter Tutorial• Deployment	Planning and Architecture <ul style="list-style-type: none">• Planning and Architecture• Understanding BizTalk Adapter for Oracle E-Business Suite• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Glossary
---	--

See Also

Concepts

[Common Developer Tasks](#)

Other Resources

[Getting Started](#)

Common Developer Tasks

The following table provides links to topics useful for developing applications on top of Microsoft BizTalk Adapter for Oracle E-Business Suite.

Developer Tasks <ul style="list-style-type: none">• Establishing a Connection to the Oracle E-Business Suite• Developing BizTalk Applications• Developing Applications By Using the WCF Service Model• Developing Applications by Using the WCF Channel Model	Planning and Architecture <ul style="list-style-type: none">• Planning and Architecture• Understanding BizTalk Adapter for Oracle E-Business Suite• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Troubleshooting• Glossary
---	--

Understanding BizTalk Adapter for Oracle E-Business Suite

The BizTalk Adapter Pack enables service-oriented programmatic access in order to interact with an external system. The adapters provide the following advantages to clients:

- **Consistent design-time experience.** The adapters provide a common and user-friendly design-time experience for browsing, searching, and retrieving metadata of LOB artifacts.
- **Varied programming options.** The adapters provide a choice of programming model including the Windows Communication Foundation (WCF) channel model, WCF service model, ADO.NET, Web services, or BizTalk Server supported models.
- **Uniform experience across LOBs.** The adapters standardize on using WCF and Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, and hence provide a uniform experience of gaining access to any LOB system.

As mentioned, the adapters are built on top of the WCF LOB Adapter SDK. This SDK provides a common basis for building integration adapters that a variety of client applications such as BizTalk Server and Microsoft Office can consume. The WCF LOB Adapter SDK aligns the adapter strategy with the Microsoft services strategy by exposing integration adapters as WCF channels. For more information about the WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation. The WCF LOB Adapter SDK documentation is installed along with the WCF LOB Adapter SDK, typically under *<installation drive>*:\Program Files\WCF LOB Adapter SDK\Documents.

To perform operations in Oracle E-Business Suite, adapter clients must have access to relevant artifacts in Oracle E-Business Suite. External applications can add or remove data in Oracle E-Business Suite interface tables and the database tables by using SQL statements. Applications can also access data in the interface tables and the database tables by using views, functions, and procedures. With Microsoft BizTalk Adapter for Oracle E-Business Suite, adapter clients can browse the artifacts in Oracle E-Business Suite as well as in the underlying database. In Oracle E-Business Suite, the adapter clients can browse interface tables, interface views, concurrent programs, and request sets while in the underlying Oracle database, the adapter clients can browse tables, views, stored procedures, functions, PL/SQL APIs, and packages. Adapter clients can select the artifacts they require for their solution and retrieve metadata for those artifacts. This enables users to access and execute the operations on the artifacts in Oracle E-Business Suite and the underlying Oracle database.

This section lists the features of the Microsoft BizTalk Adapter for Oracle E-Business Suite.

In This Section

- [Overview of BizTalk Adapter for Oracle E-Business Suite](#)
- [Key Features in BizTalk Adapter for Oracle E-Business Suite](#)
- [Limitations of BizTalk Adapter for Oracle E-Business Suite](#)

See Also

Other Resources

[Getting Started](#)

Overview of BizTalk Adapter for Oracle E-Business Suite

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes Oracle E-Business Suite as a WCF service. Adapter clients can perform operations on the Oracle E-Business Suite by exchanging SOAP messages with the adapter. The adapter consumes the SOAP message and makes appropriate ODP.NET calls to perform the operation. The adapter returns the response from Oracle E-Business Suite back to the client in the form of SOAP messages.

The Oracle E-Business adapter displays metadata of the Oracle E-Business Suite artifacts (PL/SQL APIs, interface tables/views, concurrent programs, and request sets) and the underlying Oracle database artifacts (such as tables, functions, and procedures) that describe the structure of a SOAP message in the form of Web Services Description Language (WSDL). The Oracle E-Business adapter uses the Add Adapter Service Reference Visual Studio Plug-in, Add Adapter Metadata Wizard, and Consume Adapter Service BizTalk Project Add-in to enable adapter clients to retrieve metadata for operations. The adapter also generates programming artifacts that can be used in your programming solution. For more information about Add Adapter Service Reference Visual Studio Plug-in, Add Adapter Metadata Wizard, and Consume Adapter Service BizTalk Project Add-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

The Oracle E-Business adapter uses the Oracle Data Provider for .NET (ODP.NET) 11.1.0.7 to communicate with Oracle E-Business Suite. ODP.NET 11.1.0.7 is one of the components of Oracle Data Access Components (ODAC). You can use the Oracle E-Business adapter to communicate with Oracle E-Business Suite in the following ways:

- By developing BizTalk applications. For more information, see [Developing BizTalk Applications](#).
- By using the Windows Communication Foundation (WCF) service model. For more information, see [Developing Applications By Using the WCF Service Model](#).
- By using the WCF channel model. For more information, see [Developing Applications by Using the WCF Channel Model](#).

In This Section

- [How Does the Adapter Connect to Oracle E-Business Suite?](#)
- [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#)
- [What Operations Can Be Performed Using the Adapter?](#)
- [How does the Adapter Handle Transactions?](#)
- [Other Features Supported by the Adapter](#)

See Also

Other Resources

[Understanding BizTalk Adapter for Oracle E-Business Suite](#)

How Does the Adapter Connect to Oracle E-Business Suite?

The Microsoft BizTalk Adapter for Oracle E-Business Suite uses ODP.NET 11.1.0.7 to connect to Oracle E-Business Suite. The Oracle E-Business adapter requires adapter clients to provide a connection string, called the connection Uniform Resource Identifier (URI), to connect to the Oracle E-Business Suite. Internally, the Oracle E-Business adapter connects to the underlying Oracle database through the URI. With a connection URI, adapter clients can specify connection parameters to connect to an external system.

The Oracle E-Business adapter enables adapter clients to connect to Oracle E-Business Suite in the following two ways:

- **Using tnsnames.ora:** The connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, and port number from the net service name entry in the tnsnames.ora file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.

Important

Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the [connection URI](#) cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

- **Without using tnsnames.ora:** The connection URI provided by the adapter clients contains the connection parameters such as server name, service name, and port number. In this case, the net service name in the tnsnames.ora file, or the actual tnsnames.ora file itself, does not need to be present on the client computer. This is helpful when you have a large number of users connecting to the Oracle database in your organization, and adding/updating servers does not lead to manually adding/updating the connection details in the tnsnames.ora file on every client computer.

Important

This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

For more information about connecting to Oracle E-Business Suite, see [Establishing a Connection to the Oracle E-Business Suite](#).

Make sure you comply with the security guidelines when establishing a connection with Oracle E-Business Suite. For more information about security guidelines, see [Security and Protection](#).

Specifying Client Credentials

In Oracle E-Business adapter, you can provide credentials to connect to Oracle E-Business Suite in the following two places:

- On the **Security** tab in the **Configure Adapter** dialog box. You can find this dialog box in the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in.
- In the **OracleUserName** and **OraclePassword** binding properties on the **Binding Properties** tab in the **Configure Adapter** dialog box. You can find this dialog box in the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in. In this case, the credentials are stored in plain text in the binding file.

The Oracle E-Business adapter exposes the **ClientCredentialType** binding property that allows you to specify the set of credentials (Oracle E-Business Suite or Oracle database) that will be used to connect to Oracle E-Business Suite.

- To connect using the Oracle database credentials, specify the **ClientCredentialType** binding property as **Database**, and then, on the **Security** tab, specify the database credentials in the **User name** and **Password** text boxes. If you will be performing operations on any of the Oracle E-Business Suite artifacts (interface table, interface view, concurrent program, request set, or Oracle E-Business Suite PL/SQL APIs), you must also provide the Oracle E-Business Suite credentials in the **OracleUserName** and **OraclePassword** binding properties.

- To connect using Oracle E-Business Suite credentials, specify the **ClientCredentialType** binding property as **EBusiness**, and then specify Oracle E-Business Suite credentials in the **User name** and **Password** text boxes on the **Security** tab. You must also specify the Oracle database credentials for the **OracleUserName** and **OraclePassword** binding properties.

For more information about specifying client credentials, see [Specifying Credentials for the Oracle E-Business Suite](#).

Windows Authentication

The Oracle E-Business adapter supports Windows Authentication while connecting to Oracle E-Business Suite. With Windows Authentication, the adapter clients can determine a user's identity based on Windows logon credentials, and can leverage the built-in security of the Windows environment. For information about connecting to Oracle E-Business Suite by using Windows Authentication, see [Connecting to Oracle E-Business Suite Using Windows Authentication](#).

See Also

Other Resources

[Overview of BizTalk Adapter for Oracle E-Business Suite](#)

How Does the Adapter Surface Oracle E-Business Suite Metadata?

The metadata that Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces from Oracle E-Business Suite and the underlying Oracle database describes the message structure for communicating with Oracle E-Business Suite using the adapter. The Oracle E-Business adapter supports two interfaces for retrieving metadata.

- MetadataExchange provided by Windows Communication Foundation (WCF). WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from Oracle E-Business Suite.
- IMetadataRetrievalContract provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

The Oracle E-Business adapter surfaces the Oracle E-Business Suite and the underlying database artifacts, and respective operations that the adapter clients can invoke. These operations are discussed later in this topic.

You can use the adapter clients to browse, search, and retrieve metadata by:

- Creating a BizTalk project in Visual Studio
- Using the WCF channel model
- Using the WCF service model

When using a BizTalk project, you must use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Metadata Wizard to generate metadata for the operations that you want to perform in Oracle E-Business Suite. When using the WCF service model, you must use the Add Adapter Service Reference Visual Studio Plug-in to generate the proxy classes for performing operations in Oracle E-Business Suite. For more information about browsing, searching, and retrieving metadata using Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

Browsing Metadata

The Oracle E-Business adapter enables adapter clients to browse interface tables, interface views, concurrent programs, and request sets in Oracle E-Business Suite and tables, views, stored procedures, functions, and packages in the underlying database. As part of the metadata browse operation, the adapter also surfaces the operations that can be performed on the Oracle database, including some custom operations supported by the adapters. These operations are available from Consume Adapter Service Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Plug-in.

The Oracle E-Business adapter surfaces most of the operations under the following three nodes:

1. **Application-Based View:** Contains the operations grouped by each application for the Oracle E-Business Suite artifacts.
2. **Artifact-Based View:** Contains the operations grouped by artifact type (such as Interface Tables, Interface Views, and so on) in Oracle E-Business Suite and the underlying database.
3. **Schema-Based View:** Contains the operations grouped by each schema for the underlying database artifacts.

There are some generic operations exposed at the root level that are applicable for both the nodes. Moreover, different operations are surfaced based on the type of operation: outbound or inbound.

The following table lists the outbound and inbound operations surfaced by the Oracle E-Business adapter:

Outbound Operations	Inbound Operations
---------------------	--------------------

Application-Based View:

Contains a list of Oracle Applications in the underlying Oracle E-Business Suite. Expand an Oracle Application node to see the following artifacts:

- **Interface Tables:** A list of all the interface tables. Select an interface table to view the Insert, Select, Update, and Delete operations.
- **Interface Views:** A list of all the interface views. Select an interface view to view the Select operation.
- **Concurrent Programs:** The following operations for the concurrent programs:
 - A set of all the concurrent programs specific to the Oracle Application that are exposed as operations.
 - The Get_Status operation to get the status of a concurrent program.
 - The Wait_For_Request operation to wait for a request to be completed before returning status.
 - The Submit_Request operation to call or execute a concurrent program by specifying the parameters required for the execution of the concurrent program.
- **Request Sets:** A set of all the request sets specific to the Oracle Application that are exposed as operations.

Application-Based View:

Contains a list of Oracle Applications in the underlying Oracle E-Business Suite. Expand an Oracle Application node to see the following artifacts:

- **Interface Tables:** The Poll operation for the interface tables that enables adapter clients to obtain inbound data from the Oracle E-Business Suite based on a query polling mechanism supported by the adapter.
- **Interface Views:** The Poll operation for the interface views that enables adapter clients to obtain inbound data from the Oracle E-Business Suite based on a query polling mechanism supported by the adapter.

Artifact-Based View:

Contains all the artifacts in Oracle E-Business Suite and the underlying database. Expand an artifact node to see a list of the Oracle Applications or schemas based on the origin of the artifact (applications or database). For example, the **Interface Tables** node will display a list of Oracle Applications whereas the **Tables** node will display a list of database schemas.

The **Artifact-Based View** displays the artifacts listed under **Applications-Based View** and **Schema-Based View**. Each artifact node lists the operations relevant for an Oracle Application or a database schema.

Artifact-Based View:

Except concurrent programs and request sets, contains all the artifacts in Oracle E-Business Suite and all the artifacts in the underlying database. Expand an artifact node to see a list of the Oracle Applications or schemas based on the origin of the artifact (applications or database). For example, the **Interface Tables** node will display a list of Oracle Applications whereas the **Tables** node will display a list of database schemas.

The **Artifact-Based View** displays the artifacts listed under **Applications-Based View** and **Schema-Based View**. Each artifact node lists the operations relevant for an Oracle Application or a database schema.

<p>Schema-Based View:</p> <p>Contains a list of schemas in the underlying Oracle database. Expand schema node to see the following artifacts:</p> <ul style="list-style-type: none"> • PL/SQL APIs: A list of all the PL/SQL APIs. Select a PL/SQL API to view the packaged procedures and functions that are exposed as operations. • Procedures: A list of procedures in the schema that are exposed as operations. • Functions: A list of functions in the schema that are exposed as operations. • Tables: A list of all the tables. Select a table to view the Insert, Select, Update, and Delete operations. • Views: A list of all the views. Select a view to view the Select operation. 	<p>Schema-Based View:</p> <p>Contains a list of schemas in the underlying Oracle database. Expand schema node to see the following artifacts:</p> <ul style="list-style-type: none"> • PL/SQL APIs: A list of all the PL/SQL APIs. Select a PL/SQL API to view the packaged procedures and functions that are exposed as operations for polling. • Procedures: A list of procedures in the schema that are exposed as operations for polling. • Functions: A list of functions in the schema that are exposed as operations for polling. • Tables: A list of all the tables. Select a table to view the Poll operation for the table. • Views: A list of all the views. Select a view to view the Poll operation for the view.
<p>The Oracle E-Business adapter also exposes the following generic outbound operations at the root level: ExecuteReader, ExecuteScalar, and ExecuteNonQuery. For information about these operations, see Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations.</p>	<p>The Oracle E-Business adapter also exposes the Notification operation at the root level that enables adapter clients to receive database change notification messages from Oracle E-Business Suite. For more information about the notification operation, see Receiving Database Change Notifications.</p>

For more information about how the metadata is categorized, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

Searching Metadata

Using the Oracle E-Business adapter, you can perform a search query in Oracle E-Business Suite and on the underlying Oracle database by using the Oracle search expressions that are compatible with the LIKE operator. For example, adapter clients can use a search expression such as "EMP%" to obtain tables starting with EMP. The adapter converts this to the following SQL query:

```
SELECT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME LIKE 'EMP%' AND OWNER = 'SCOTT'
```

Where, SCOTT is the schema with a collection of Oracle database artifacts.

The following table lists the special characters that can be used for search and their interpretation by the Oracle Database adapter.

Special character	Interpretation
_ (underscore)	Matches exactly one character For example, A_ matches AB, AC, and AD.
% (percent sign)	Matches zero or more characters. For example, A% matches A, AB, ABC.

\ (escape)	Escapes the special meaning of % and _. The \ (escape) character is used before a wildcard character to indicate that the wildcard character should be interpreted as a regular character. For example, A_B matches A_B.
------------	--

◆ Important

- The search string is case-sensitive.
- The search works differently under the different view (Application-Based View, Artifact-Based View, and Schema-Based View). To know how you can search for artifacts and operations under each view, see "Searching Under Different Views" in [Searching for Operations](#).
- To search for an application you can specify the friendly name or the short name of the application. For example, to search for the **Receivables** application you can specify the search string as either **Receive%** or **AR**. AR is the application's short name.
- To search for a concurrent program you can specify the friendly name or the actual name of the concurrent program. For example, to search for the **Customer Interface** concurrent program you can specify the search string as either **%Customer Interface%** or **%RACUST%**. RACUST is the actual name of the concurrent program. Also, the search result will always contain the standard concurrent programs irrespective of whether their name matches with the specified search string.

Retrieving Metadata

When retrieving metadata, the Oracle E-Business adapter can extract metadata under a schema, including all or a subset of database objects with the respective object and operation parameters. The adapter presents the entities from Oracle E-Business Suite and the underlying Oracle database as element names in XML. Because underscores are the only permissible special characters that can be included, all other special characters in the element names are encoded using underscores. For example, emp\$name is encoded as emp_x0024_name. For more information, see [Retrieving Metadata for Operations](#).

See Also

Other Resources

[Overview of BizTalk Adapter for Oracle E-Business Suite](#)

[Understanding BizTalk Adapter for Oracle E-Business Suite](#)

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

[Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#)

What Operations Can Be Performed Using the Adapter?

Adapter clients can perform operations in Oracle E-Business Suite by:

- Creating BizTalk projects
- Using the WCF channel model
- Using the WCF service model

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel. For information about the message structure and the SOAP action associated with each operation, see [Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#).

This section provides information about the operations supported in Oracle E-Business Suite using the Oracle E-Business adapter.

In This Section

- [Setting Application Context](#)
- [Operations on Interface Tables and Interface Views](#)
- [Operations on PL/SQL APIs](#)
- [Operations on Concurrent Programs](#)
- [Operations on Request Sets](#)
- [Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#)
- [Operations on Tables That Contain BFILE Data Types](#)
- [Operations on Functions and Stored Procedures](#)
- [Operations on Functions and Procedures with REF CURSOR Parameters](#)
- [Operations on Functions and Procedures with RECORD Types](#)
- [Operations on Synonyms](#)
- [Receiving Database Change Notifications](#)
- [Support for Inbound Calls Using Polling](#)
- [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#)
- [Support for Composite Operations](#)
- [Support for Oracle User-Defined Types](#)

See Also

Other Resources

[Overview of BizTalk Adapter for Oracle E-Business Suite](#)

Setting Application Context

In Microsoft BizTalk Adapter for Oracle E-Business Suite, setting application context is mandatory for some Oracle E-Business Suite artifacts (interface tables, interface views, concurrent programs, and request sets) before you can perform operations on them. The Oracle E-Business adapter does not allow you to perform operations on these artifacts until you have set the application context. However, for artifacts in the underlying Oracle database, it is up to the user whether they want to set the application context or not.

What is Application Context

Application context is a set of elements associated with an artifact in Oracle E-Business Suite that implements user preferences and access control on the artifact. Application context consists of the following elements:

- **User name:** A user that can connect to Oracle E-Business Suite.
- **Responsibility:** A responsibility is an access level in Oracle E-Business Suite that allows users to access only those data and functions that are appropriate to their roles in an organization. Responsibilities can allow access to a specific application, operating units, set of books, and a restricted list of windows, functions, and other responsibilities. By virtue of assigning responsibilities to a user, you can grant/restrict access of the user in Oracle E-Business Suite.
- **Organization ID:** Oracle E-Business Suite supports setting up of multiple organizations. These different organizations are uniquely identified by a value, Organization ID, in the Org_ID column of the table in Oracle E-Business Suite that stores information about these organizations. By virtue of assigning a responsibility to an organization or selecting an organization explicitly, you can grant/restrict access of a user to an organization.

For more information about responsibility, multiple organizations, and Organization ID in Oracle E-Business Suite, consult the Oracle documentation at <http://go.microsoft.com/fwlink/?LinkId=124538>.

Setting Application Context

As the Oracle E-Business adapter connects to the underlying database in the Oracle E-Business Suite, application context for the Oracle E-Business Suite artifacts are not established or initialized in the adapter. You can initialize or set the application context for these artifacts in the Oracle E-Business adapter by using either of the following:

- **Binding properties:** The Oracle E-Business adapter exposes the following binding properties for setting the application context: **OracleEBSOrganizationId**, **OracleUserName**, **OraclePassword**, **OracleEBSResponsibilityKey**, **OracleEBSResponsibilityName**, and **ApplicationShortName**. You do not need to specify values for all these binding properties to set application context for various artifacts. For information about the binding properties required for setting application context for an artifact, see [Binding Properties for Setting Application Context](#) later in this topic.
- **Message context properties:** The Oracle E-Business adapter exposes the following message context properties for setting the application context: **ApplicationShortName**, **OrganizationID**, **ResponsibilityKey**, and **ResponsibilityName**. For specifying the user name and password, you must use the binding properties. For information about how to set application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

Important

The value specified for the **OracleEBSResponsibilityKey** binding property overrides the value of the **OracleEBSResponsibilityName** binding property. Similarly, the value specified for the **ResponsibilityKey** message context property overrides the value specified for the **ResponsibilityName** message context property.

Precedence Order (Binding Properties vs. Message Context Properties)

If you set the application context using both the binding properties and message context properties, the values specified for message context properties takes precedence and overrides the values specified for the binding properties. But, for example, if you specify the application short name as a message context property and the others as binding properties, only the value for the application short name is taken from the message context property and the rest are picked from the relevant binding properties.

Precedence Order for Application Short Name

While setting the application context, the application short name is used in the following precedence order (highest to lowest):

- The application short name specified in the **ApplicationShortName** message context property.
- The application short name specified in the SOAP action (for interface tables, interface views, concurrent programs, and request sets only).
- The application short name specified in the **ApplicationShortName** binding property.

However, for interface tables, interface views, concurrent programs, and request sets, this precedence order is only applicable while setting the application context. To identify the interface tables, interface views, concurrent programs, and request sets, the application short name in the SOAP action is used.

Precedence Order for Responsibility Key and Responsibility Name

While setting the application context, the responsibility key and responsibility name are used in the following precedence order (highest to lowest):

- The responsibility key specified in the **ResponsibilityKey** message context property.
- The responsibility name specified in the **ResponsibilityName** message context property.
- The responsibility key specified in the **OracleEBSResponsibilityKey** binding property.
- The responsibility name specified in the **OracleEBSResponsibilityName** binding property.

Tip

Why use message context properties over binding properties to set the application context? If you set the application context using binding properties, the WCF-Custom send port for the Oracle E-Business adapter can be used only for the specific organization ID, responsibility, and application that you specified for the binding properties. On the contrary, if you use the message context property you can configure a "generic" WCF-Custom send port and set the application context at the message level.

Setting Application Context For Interface Tables, Interface Views, Concurrent Programs, and Request Sets (Mandatory)

You must set the application context before performing operations on interface tables, interface views, concurrent programs, and request sets in Oracle E-Business adapter. To do this, you have to provide appropriate values for the binding properties or the message context properties as specified earlier.

Important

You cannot perform operations on interface tables, interface views, concurrent programs, and request sets unless you have set appropriate values for the required binding properties or the message context properties.

Setting Application Context For PL/SQL APIs, Procedures, Functions, Tables, and Views

- **PL/SQL APIs:** The Oracle E-Business adapter exposes PL/SQL APIs associated with the Oracle database as well as the Oracle E-Business Suite application. While it is optional to set the application context for the PL/SQL APIs associated with the Oracle database, it is mandatory to set the application context for the PL/SQL APIs associated with the Oracle E-Business Suite application.
- **Procedures and Functions:** It is not mandatory to set the application context to perform operations on procedures and functions in the Oracle database.
- **Tables and Views:** It is not mandatory to set the application context to perform operations on tables and views in the Oracle database. However, for custom Oracle E-Business Suite application, users may or may not register the base database tables as interface tables. If a database table is not registered as an interface table, it will be displayed along with the database tables in the Oracle E-Business adapter. Because these tables are associated with an Oracle E-Business

application, for any operation on these tables you must set the application context.

To set the application context for these artifacts, you must provide appropriate values for the binding properties or the message context properties as specified earlier.

Setting Application Context For Poll, ExecuteNonQuery, ExecuteReader, ExecuteScalar, and Composite Operations

Apart from the artifacts, you can also set the application context for various operations that are performed on these artifacts.

- To set the application context for the Poll operation, you can only use the binding properties as specified earlier. For setting application context, you must provide appropriate values for the binding properties that are applicable for the artifact on which the Poll operation is performed. For example, if the Poll operation is performed on an interface table then you must specify values for the binding properties for the interface table.
- To set the application context for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations, you must provide appropriate values for the binding properties or the message context properties as specified earlier. For setting application context for these operations, you must provide appropriate values for the binding properties or the message context properties that are applicable for the artifact on which the operations are performed.
- To set the application context for composite operations, you must provide appropriate values for the binding properties or the message context properties as specified earlier. For setting application context for composite operations, you must provide appropriate values for the binding properties or the message context properties that are applicable for the individual operations. For example, if a composite operation contains two operations: one on the interface table and the other on the database table then you must specify values for the binding properties or the message context properties for the interface table as well as the binding properties or the message context properties for the database table.

◆ Important
For all these operations, it is mandatory to set the application context if the operation is performed on an artifact in Oracle E-Business Suite (interface table, interface view, concurrent programs or request sets). If the operation is performed on an artifact in the underlying database, it is not mandatory to set the application context. For example, if you are performing the Poll operation on an interface table, it is mandatory to set the application context whereas if the Poll operation is performed on a table, it is not mandatory to set the application context.

Setting the Language for Performing Operations

The Oracle E-Business adapter supports the Multi-Language Support (MLS) feature of Oracle E-Business Suite, and allows you to specify a language while performing operations. The adapter exposes the **Language** binding property under the **MlsSettings** binding property and the **Language** message context property to specify a language for performing operations.

The value specified for the **Language** message context property overrides the value of the **Language** binding property under the **MlsSettings** binding property. For more information about the **MlsSettings** binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Binding Properties for Setting Application Context For Various Artifacts

The following table provides information about the binding properties for which you must specify appropriate values to set application context for various artifacts:

Artifacts	OracleEBSOrganizationId	OracleUserName	OraclePassword	OracleEBSResponsibilityKey or OracleEBSResponsibilityName	ApplicationShort Name
Interface Tables and Interface Views	√*	√	√	√	
Concurrent Programs	√*	√	√	√	
Request Sets	√*	√	√	√	

PL/SQL APIs	√*	√	√	√	√
Procedures and Functions	√*	√	√	√	√
Tables and Views	√*	√	√	√	√

√* = **Optional**

◆ Important

- The default value of the **OracleEBSOrganizationId** binding property (optional) is null. If you specify a value for the **OracleEBSOrganizationId** binding property, the Oracle E-Business adapter sets the ORG_ID of the session to this value while setting the application context.
- The value specified for the **OracleEBSResponsibilityKey** binding property overrides the value specified for the **OracleEBSResponsibilityName** binding property.

For detailed information about each of these binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Interface Tables and Interface Views

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces a set of standard operations (Select, Insert, Update, and Delete) for each interface table, and the Select operation for each interface view in Oracle E-Business Suite. By using these operations, you can perform the SELECT, INSERT, UPDATE, and DELETE statements qualified by a WHERE clause on the target interface table, and the SELECT statement qualified by a WHERE clause on the target interface view. These operations are also called data manipulation language (DML) operations.

Important

Before you can perform operations on interface tables and interface views, you must set the applications context for these artifacts in Oracle E-Business adapter. This is because setting applications context facilitates secure transactions in Oracle E-Business Suite by setting user preferences (such as responsibility, organization, and language settings) and access control for an artifact. For more information about applications context and how to set it, see [Setting Application Context](#).

The following table shows the DML operations that the Oracle E-Business adapter supports:

Operation	Description
Select	<p>Performs a Select Operation on the target interface table or interface view based on a supplied list of column names and a filter string that specifies a SQL WHERE clause.</p> <p>The return value for a Select operation is a strongly-typed result set that contains the specified columns and rows.</p>

In Performs an Insert operation on the target interface table. The Insert operation supports multiple-record insert into the target interface table based on a supplied record set.

rt The return value for an Insert operation is the number of rows inserted.

InlineValue

For all simple data records in an Insert operation, you can choose to override the value of a record by specifying a value for an optional attribute called **InlineValue**. The InlineValue attribute can be used to insert computed values into interface tables such as populating the primary key column using a sequence or inserting system date (using SYSDATE) into a date column. For example, in the following INSERT statement:

```
<Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/AR/AR_ARCHIVE_PURGE_INTERIM">
  <RECORDSET>
    <InsertRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecord/AR/AR_ARCHIVE_PURGE_INTERIM">
      <TRNS_DATE InlineValue="sysdate">2008-06-21T15:52:19</TRNS_DATE>
      <EMPNAME>John</EMPNAME>
    </InsertRecord>
  </RECORDSET>
</Insert>
```

Even though "2008-06-21T15:52:19" is specified as a value for TRNS_DATE, the value of the **InlineValue** attribute, "SYSDATE," (system date) will be inserted into the target interface table.

While using the InlineValue attribute:

- Avoid using constant values for the InlineValue attribute. For example, in the INSERT statement, if you specify `<EMPNAME InlineValue="John"/>` then it will result in an error. This is because the value of the InlineValue attribute is passed on as-is to Oracle, and in this case *John* is passed to Oracle E-Business Suite, which is not the expected value (expected value is *John*). You would have to use single quotes around the employee name. For example: `<EMPNAME InlineValue="' John'"/>`.
- If you want to use a select query for the InlineValue attribute, you must enclose the SELECT statement in parentheses and also ensure that the select query fetches only a single record. For example: `<EMPNAME InlineValue="(SELECT NAME FROM MS_SAMPLE_EMPLOYEES WHERE ID=123)"/>`.

Note

If an element is marked as NOT NULL in Oracle E-Business Suite, you must specify a value for that element even if you have specified an inline value. Failing to do this will cause the schema validation to fail.

U Performs an Update operation on the target interface table. The records to be updated are specified by a filter string that specifies a SQL WHERE clause. The values for the update are specified in a template record.

da The return value for an Update operation is the number of rows updated.

D Performs a Delete operation on the target interface table based on a SQL WHERE clause that is specified in a filter string.

et The return value for a Delete operation is the number of rows deleted.

Important

- The Oracle E-Business adapter surfaces the same set of standard operations (Select, Insert, Update, and Delete) for each table, and the Select operation for each view in the underlying Oracle database. The above DML operations are also valid for the underlying Oracle database tables and views.
- It is not mandatory to set the applications context to perform operations on tables and views in the Oracle database. However, for custom Oracle E-Business Suite applications, users may or may not register the base database tables as interface tables. If a database table is not registered as an interface table, it will be available under the **Tables** subnode in the **Artifact-Based View** node or in the **Schema Based View** node at design-time while using Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in. Because these tables are associated with an Oracle E-Business application, for any operation on these tables you must set the application context. For more information about applications context and how to set it, see [Setting Application Context](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on PL/SQL APIs

Oracle E-Business Suite provides a set of PL/SQL APIs in the form of packaged functions and stored procedures. These packaged functions and procedures are surfaced as operations in Microsoft BizTalk Adapter for Oracle E-Business Suite. The PL/SQL APIs are grouped by schema names under the **Artifact-Based View** and **Schema-Based View** nodes when you connect to Oracle E-Business Suite using Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in. For information about browsing the PL/SQL APIs in the Oracle E-Business adapter, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

The Oracle E-Business adapter displays the PL/SQL APIs associated with the Oracle database as well as the applications in Oracle E-Business Suite under the **Artifact-Based View** and **Schema-Based View** nodes.

◆ Important

- Before you can perform operations on PL/SQL APIs associated with applications in Oracle E-Business Suite, you must set the applications context. This is because setting applications context facilitates secure transactions in Oracle E-Business Suite by setting user preferences (such as responsibility, organization, and language settings) and access control for an artifact. However, it is optional to set the applications context for PL/SQL APIs associated with Oracle database. For information about applications context, and how to set it, see [Setting Application Context](#).
- The Oracle E-Business adapter cannot ascertain whether or not a default value is assigned for a parameter in a PL/SQL API in Oracle. Moreover, the adapter also cannot ascertain whether a parameter is defined as mandatory or optional in a PL/SQL API in Oracle. The adapter treats every parameter as an optional parameter, and if no value is specified for a parameter that:
 - Has a default value specified in Oracle then the default value is used.
 - Is defined as a mandatory parameter in Oracle then an exception is thrown.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Concurrent Programs

Concurrent programs in Oracle E-Business Suite are surfaced as operations in Microsoft BizTalk Adapter for Oracle E-Business Suite. Along with the concurrent programs specific to an Oracle application, the Oracle E-Business adapter also surfaces the following three standard operations: `Get_Status`, `Wait_For_Request`, and `Submit_Request`. This implies that if an Oracle application has two concurrent programs, five operations will be exposed: one for each concurrent program, and three for the standard operations.

For information about:

- Browsing and searching concurrent programs, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).
- How to invoke concurrent programs in the Oracle E-Business adapter, see [Invoking Concurrent Programs in Oracle E-Business Suite Using BizTalk Server](#).

Important

You must set the applications context for concurrent programs in Oracle E-Business adapter before you can perform any operations on concurrent programs. This is because setting applications context facilitates secure transactions in Oracle E-Business Suite by setting user preferences (such as responsibility, organization, and language settings) and access control for an artifact. For information about applications context, and how to set it, see [Setting Application Context](#).

The following sections provide information about the operations exposed by the Oracle E-Business adapter for concurrent programs.

<Concurrent_Program_Name> Operation

As mentioned earlier, there will be as many <Concurrent_Program_Name> operations as the number of concurrent programs in an Oracle Application. The <Concurrent_Program_Name> operation takes five standard parameters: three of complex type and two of simple type.

Note

For the concurrent programs that do not expose their metadata, the Oracle E-Business adapter exposes 100 optional parameters for each of these concurrent programs. To invoke these concurrent programs successfully, the user must consult the Oracle E-Business Suite documentation to figure out the parameters for a concurrent program that require a value, and then specify them. An example of such a concurrent program is **Journal Import** (actual name: **GLLEZL**) in the **General Ledger** application.

Complex type parameters

- **SetOptions:** Enables you to set options for the concurrent program before submitting the request. `SetOptions` takes the following options as parameters:
 - **Implicit:** Indicates whether to display the concurrent request in the user's Concurrent Requests form in Oracle E-Business Suite. You can specify any of the following four values: **No**, **Yes**, **Error** or **Warning**. Specifying **No** causes the requests to be displayed in the user's Concurrent Requests form in Oracle E-Business Suite. Specifying **Yes** implies that the request may be viewed only from the system administrator's privileged Concurrent Requests form. Specifying **Error** causes the request to be displayed in the user's Concurrent Requests form only if it fails. Specifying **Warning** causes the request to display in the user's Concurrent Requests form only if there is a warning or an error.
 - **Protected:** Indicates whether the concurrent request is protected against updates made using the Concurrent Requests form in Oracle E-Business Suite. You can specify **Yes** (protected) or **No** (not protected).
 - **Language:** Indicates the National Language Support (NLS) language. If no value is specified, it defaults to the current language.

- **Territory:** Indicates the language territory. If no value is specified, it defaults to the current language territory.
- **ContinueOnFail:** Indicates whether the concurrent request submission should continue or throw an exception in case **SetOptions** fails. You can specify **True** (continue) or **False** (throw an exception).
- **SetPrintOptions:** Enables you to set the print options for the concurrent program before submitting the request. **SetPrintOptions** takes the following options as parameters:
 - **Printer:** Indicates the printer name where the concurrent request output should be sent. You cannot override this print option if it is already set in the Concurrent Programs form in Oracle E-Business Suite.
 - **Style:** Indicates the print style used to print the concurrent request output. For example, you can specify the orientation (**Landscape** or **Portrait**). If the print style is already set in the Concurrent Programs form in Oracle E-Business Suite, and the **Style Required** check box is selected, you cannot override this print option.
 - **Copies:** Indicates the number of copies to be printed of the concurrent request output.
 - **SaveOutput:** Indicates whether or not to save the output file. You can specify **Yes** or **No**.
 - **PrintTogether:** Applicable only for those requests that contain sub-requests. Indicates how the output of sub-requests is printed. If you specify **Y**, the output of sub-requests is printed only after all the sub-requests are complete. If you specify **N**, the output of each sub-request is printed as it completes.
 - **ContinueOnFail:** Indicates whether the concurrent request submission should continue or throw an exception in case **SetPrintOptions** fails. You can specify **True** (continue) or **False** (throw an exception).
- **SetRepeatOptions:** Enables you to set the repeat options for the concurrent program before submitting the request. **SetRepeatOptions** takes the following options as parameters:
 - **RepeatTime:** Indicates the time of day to repeat the concurrent request.
 - **RepeatInterval:** This parameter is applicable only when **RepeatTime** is NULL. Indicates the interval between resubmissions of the request. Use this option along with **RepeatUnit** to specify the time between resubmissions.
 - **RepeatUnit:** This parameter is applicable only when **RepeatTime** is NULL. The unit of time used along with **RepeatInterval** to specify the time between resubmissions of the request. You can specify **Minutes**, **Hours**, **Days** or **Months**.
 - **RepeatType:** This parameter is applicable only when **RepeatTime** is NULL. Indicates whether the repeat interval is applied after the "start" of a concurrent request execution or after the "end" of a concurrent request execution.
 - **RepeatEndTime:** Indicates the date and time to stop resubmitting the concurrent request.
 - **ContinueOnFail:** Indicates whether the concurrent request submission should continue or throw an exception in case **SetRepeatOptions**. You can specify **True** (continue) or **False** (throw an exception).

Simple Type Parameters

- **Description:** Description of the concurrent request.
- **StartTime:** Indicates the time at which the concurrent request should start running.

Get_Status operation

The standard operation, **Get_Status**, returns the request phase/status and the completion message of a concurrent program.

This operation takes the request ID of a concurrent program (**RequestID**) as an input, and then returns the following information:

- **Phase:** The user-friendly request phase from FND_LOOKUPS.
- **Status:** The user-friendly request status from FND_LOOKUPS.
- **DevPhase:** The request phase as a string that can be used for program logic comparisons.
- **DevStatus:** The request status as a string that can be used for program logic comparisons.
- **Message:** The completion message if the request has completed.

Wait_For_Request operation

The standard operation, `Wait_For_Request`, waits for request completion, and then returns the request phase/status and the completion message. This operation takes the request ID of a concurrent program (**RequestID**), the number of seconds to wait between checks (**Interval**), and the maximum time in seconds to wait for the request's completion (**MaxWait**) as input parameters, and then returns the same information as in the `Get_Status` operation.

Submit_Request operation

The standard operation, `Submit_Request`, submits a concurrent request for processing by a concurrent manager. If the request completes successfully, this operation returns the concurrent request ID. Otherwise, it returns "0".

The `Submit_Request` operation takes six standard parameters: three each of complex type simple type. Apart from these parameters, it also takes the arguments of the concurrent program as an array of string.

Complex type parameters

The `Submit_Request` operation takes **SetOptions**, **SetPrintOptions**, and **SetRepeatOptions** as input parameters. For information about these parameters, see [<Concurrent_Program_Name> Operation](#) earlier in this section.

Simple type parameters

- **Program:** Short name of the concurrent program for which the request should be submitted.
- **Description:** Description of the concurrent request.
- **StartTime:** The time at which the concurrent request should start running.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Request Sets

A request set in Oracle E-Business Suite is a set of reports and concurrent programs that are organized into various stages. You can use a single request set to run a set of reports and concurrent programs. Request sets are divided into one or more stages, and each stage contains a set of reports and concurrent programs. These stages are linked with each other, and the order of the execution of each stage is defined. For more information about request sets, see <http://go.microsoft.com/fwlink/?LinkId=129539>.

Microsoft BizTalk Adapter for Oracle E-Business Suite enables you to execute request sets in Oracle E-Business Suite. The request sets are exposed as operations in the Oracle E-Business adapter. Since a request set contains a set of concurrent programs, those concurrent programs are the input parameters for a request set operation. Along with the concurrent programs, the request set operation takes four complex type parameters and a simple type parameter as input.

◆ Important

You must set the applications context for request sets in Oracle E-Business adapter before you can perform any operations on request sets. This is because setting applications context facilitates secure transactions in Oracle E-Business Suite by setting user preferences (such as responsibility, organization, and language settings) and access control for an artifact. For information about applications context, and how to set it, see [Setting Application Context](#).

Complex type parameters

- **SetRelClassOptions:** Enables you to set scheduling options for the request set. If both SetRelClassOptions and SetRepeatOptions are set then SetRelClassOptions will take precedence. SetRelClassOptions takes the following options as parameters:
 - **Application:** Indicates the short name of the application associated with the request set.
 - **ClassName:** Indicates the name of the class associated with the request set.
 - **CancelOrHold:** Indicates the Cancel or Hold flag.
 - **StaleDate:** Indicates the date on or after which this request set will be canceled if the request set has not yet run.
 - **ContinueOnFail:** Indicates whether the request set submission should continue or throw an exception in case SetRelClassOptions fails. You can specify "True" (continue) or "False" (throw an exception).
- **SetPrintOptions:** Enables you to set the print options for the request set. SetPrintOptions takes the following options as parameters:
 - **Printer:** Indicates the printer name where the request set output should be sent.
 - **Style:** Indicates the print style used to print the request set output. For example, you can specify the orientation ("Landscape" or "Portrait").
 - **Copies:** Indicates the number of copies to be printed of the request set output.
 - **SaveOutput:** Indicates whether or not to save the output file. You can specify "True" or "False".
 - **PrintTogether:** Applicable only for sub-requests. Indicates how the output of sub-requests is printed. If you specify "Y", the output of sub-requests is printed only after all the sub-requests are complete. If you specify "N", the output of each sub-request is printed as it completes.
 - **ContinueOnFail:** Indicates whether the request set submission should continue or throw an exception in case SetPrintOptions fails. You can specify "True" (continue) or "False" (throw an exception).
- **SetRepeatOptions:** Enables you to set the repeat options for the request set. SetRepeatOptions takes the following

options as parameters:

- **RepeatTime**: Indicates the time of day to repeat the request set.
- **RepeatInterval**: This parameter is applicable only when **RepeatTime** is NULL. Indicates the interval between resubmissions of the request. Use this option along with **RepeatUnit** to specify the time between resubmissions.
- **RepeatUnit**: This parameter is applicable only when **RepeatTime** is NULL. The unit of time used along with **RepeatInterval** to specify the time between resubmissions of the request. You can specify "Minutes", "Hours", "Days" or "Months".
- **RepeatType**: This parameter is applicable only when **RepeatTime** is NULL. Indicates whether the repeat interval is applied after the "start" of a previous request set execution or after the "end" of a previous request set execution.
- **RepeatEndTime**: Indicates the date and time to stop resubmitting the request set.
- **ContinueOnFail**: Indicates whether the request set submission should continue or throw an exception in case SetRepeatOptions fails. You can specify "True" (continue) or "False" (throw an exception).
- **SetNlsOptions**: Enables you to set the NLS options for the request set. SetNlsOptions takes the following options as parameters:
 - **Language**: Indicates the NLS language.
 - **Language**: Indicates the language territory.
 - **ContinueOnFail**: Indicates whether the request set submission should continue or throw an exception in case SetNlsOptions fails. You can specify "True" (continue) or "False" (throw an exception).

Simple type parameter

StartTime: Indicates the time at which the request set should start running.

If the request set completes successfully, a concurrent request ID is returned. Otherwise, "0" is returned.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data

The Microsoft BizTalk Adapter for Oracle E-Business Suite provides support for the Oracle large object (LOB) data types:

- Binary large object (BLOB)
- Character large object (CLOB)
- National character large object (NCLOB)
- Binary file (BFILE). For more information, see [Operations on Tables That Contain BFILE Data Types](#).

In the underlying Oracle database, LOB data types are used to store large amounts of data, up to 4 gigabytes (GB). Except for the BFILE data type, LOB data types support both input and output streaming.

The Oracle E-Business adapter surfaces the following operations for interface tables, interface views, tables and views that contain LOB columns:

- **Read_<LOBColName>**: The Read_<LOBColName> operation is surfaced for interface tables, interface views, tables and views that contain BLOB, CLOB, NCLOB, and BFILE columns, where <LOBColName> is the name of the column of type BLOB, CLOB, NCLOB or BFILE. By using the Read_<LOBColName> operation, adapter clients can read values in an LOB column as a data stream. This operation takes a filter string as parameter.

Note

The Read_<LOBColName> operation is designed to support input streaming of LOB data in the WCF service model. You should use a table Select operation to read LOB data from a WCF channel model or BizTalk Server solution.

- **Update_<LOBColName>**: The Update_<LOBColName> operation is surfaced for interface tables and tables only that contain BLOB, CLOB, and NCLOB columns, where <LOBColName> is the name of the column of type BLOB, CLOB, and NCLOB. By using the Update_<LOBColName> operation, adapter clients can update values in an LOB column. For the BLOB data type, this operation takes base64binary encoded data as the parameter, whereas for the CLOB and NCLOB data types, this operation takes a string filter as the parameter.

Note

The Update_<LOBColName> operation:

- Is not supported for the BFILE data type. Adapter clients can alternatively use the Update operation. For more information, see [Operations on Tables That Contain BFILE Data Types](#).
- Is not exposed for interface views and views.
- Must be performed as part of a transaction. To ensure this, the **UseAmbientTransaction** binding property must be set to **True**. For information about the **UseAmbientTransaction** binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Important

The Oracle E-Business adapter surfaces a Read_<LOBColName> and an Update_<LOBColName> operation for each LOB column in a table. So, if there are two LOB columns in a table (LOBCol1 and LOBCol2), you will have two Read_<LOBColName> operations (Read_LOBCol1 and Read_LOBCol2) and two Update_<LOBColName> operations (Update_LOBCol1 and Update_LOBCol2).

For more information about:

- Invoking the Read_<LOBColumnName> and Update_<LOBColumnName> operations on a table in the underlying database in Oracle E-Business Suite using BizTalk Server, see [Performing Operations on Tables with Large Data Types Using BizTalk Server](#).
- Message structure and SOAP actions for performing Read_<LOBColumnName> and Update_<LOBColumnName> operations, see [Message Schemas for Special LOB Operations](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Tables That Contain BFILE Data Types

The Microsoft BizTalk Adapter for Oracle E-Business Suite supports the BFILE data type in tables and stored procedures. The following table summarizes the BFILE data type exposed by the adapter based on the operation performed and the LOB artifact (table/procedure) accessed:

Artifact	Operation	Data type exposed for BFILE column/param	Comments
TABLE	INSERT	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
TABLE	UPDATE	String	Represents the logical Oracle directory path to the file to be updated into the BFILE column
TABLE	SELECT	byte[]	Represents the binary data constituting the BFILE
STORED PROC	IN PARAM	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
STORED PROC	OUT PARAM	String	Represents the logical Oracle directory path to the file to be inserted into the BFILE column E.g. MYDIR/screen.jpg where MYDIR is a logical directory in Oracle
STORED PROC	INOUT PARAM	Not Supported	-

The special operation `Read_<LOBColumnName>` is also supported on tables with BFILE data type, where `<LOBColumnName>` is the LOB column name in the table. The `Update_<LOBColumnName>` operation is not supported for BFILE data type. Adapter clients can alternately use the Update operation.

For information about:

- The `Read_<LOBColumnName>` and `Update_<LOBColumnName>` operations, see [Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Functions and Stored Procedures

The Microsoft BizTalk Adapter for Oracle E-Business Suite supports Oracle functions and procedures in the following manner:

- **Functions** are surfaced as operations. The name of the operation is the name of the Oracle function. IN, OUT, and IN OUT parameters are supported, as well as, RETURN values.

Important

If you pass an invalid parameter in a function (that is, pass a string value for a numeric field), the Oracle E-Business adapter might throw an exception depending on the ODP.NET behavior. This is because the Oracle E-Business adapter uses ODP.NET to communicate with Oracle E-Business Suite.

- **Procedures** are surfaced as operations. The name of the operation is the name of the Oracle procedure. IN, OUT, and IN OUT parameters are supported.

Important

As part of a procedure, if you insert or update a decimal value (for example, 15.2) into a numeric field of an interface table or database table, the Oracle E-Business adapter will throw an exception. This is because the Oracle E-Business adapter uses ODP.NET to communicate with Oracle E-Business Suite, and ODP.NET does not support accepting decimal values for the numeric fields.

- **REF CURSOR types** are supported for IN and OUT parameters for procedures and functions, as well as for function RETURN values. For more information, see [Operations on Functions and Procedures with REF CURSOR Parameters](#).
- **RECORD types** are supported for IN, OUT, and IN OUT parameters for procedures and functions, as well as for function RETURN values. Both simple and complex (nested) RECORD types are supported. [Operations on Functions and Procedures with RECORD Types](#)

Note

You can also set the applications context for functions and stored procedures in Oracle E-Business adapter. For information about applications context, and how to set it, see [Setting Application Context](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Functions and Procedures with REF CURSOR Parameters

A REF CURSOR is a PL/SQL data type that represents a pointer to a server-side result set generated by executing a query. A REF CURSOR type enables input and output streaming of data and is ideal for transferring large amounts of data to and from a PL/SQL code. The Microsoft BizTalk Adapter for Oracle E-Business Suite provides support for strongly-typed and weakly-typed (SYS_REFCURSOR) REF CURSORs that can be passed to PL/SQL procedures and functions as IN and OUT parameters.

- **IN REF CURSOR.** Adapter clients must use an IN REF CURSOR by supplying a PL/SQL code (as string) that opens REF CURSOR on the Oracle database. The adapter creates variable and sets it the opened REF CURSOR and calls a function or procedure with that variable. Therefore, IN REF CURSOR parameters in PL/SQL stored procedure and functions should be represented as strings that take a PL/SQL code block as input value marking the OUT REF CURSOR variable with a "?".
- **OUT REF CURSOR.** OUT REF CURSOR parameters are returned as either strongly-typed or weakly-typed result sets. The type of the result set returned depends on whether the REF CURSOR parameter is declared as a strongly-typed or weakly-typed REF CURSOR in the stored procedure or function definition on the Oracle server.
- **IN OUT REF CURSOR parameters.** Because the Oracle E-Business adapter models IN REF CURSOR parameters as strings and OUT REF CURSOR parameters as complex types, it cannot support a single type for an IN OUT REF CURSOR parameter. For this reason, it treats IN OUT REF CURSOR parameters as two different parameters: an IN parameter in the request message and an OUT parameter in the response message.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Functions and Procedures with RECORD Types

Oracle RECORD types are used to represent hierarchical information in parameters passed to PL/SQL functions and procedures. The Microsoft BizTalk Adapter for Oracle Database surfaces RECORD types as complex XML types. The Microsoft BizTalk Adapter for Oracle E-Business Suite supports the following kinds of RECORD types:

- RECORD types that are declared as TABLE%ROWTYPE parameters in stored procedures and functions.
- RECORD types that are declared as TYPE of RECORD parameters in PL/SQL packages. For example, TYPE rec_type1 IS RECORD(name varchar2(100), age number(3));
- RECORD types that contain nested records.
- RECORD types that appear as IN, OUT, or IN OUT parameters to procedures or functions.
- RECORD types that are RETURN values of functions.

 **Note**

The Oracle E-Business adapter does not support BFILE types as RECORD members.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Synonyms

The Oracle E-Business adapter allows you to perform operations on synonyms. A synonym is an alias or friendly name for the database objects (such as tables, views, stored procedures, functions, and packages). For more information about synonyms in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=138058>.

Advantages of Using Synonyms

Synonyms are helpful in the following scenarios:

- **Working with different schemas:** If you are working with different schemas, and need to access the objects across schemas, you have to use different SQL statements to access those objects. You can create a synonym for an object in a schema, and use the synonym in your SQL statement to access the object. If you need to access the underlying object in a different schema, modify the definition of the synonym to point to the object in a different schema. Thus, the applications based on the synonym continue to function without modification in the SQL statement.

For example, suppose you have two identical schemas for your test and production environments: "Test" and "Prod." To access a table called "Employee" in the "Test" schema, you must use `Test.Employee` or `Employee` (if "Test" is the default schema) in your SQL statement. If you want to use the "Employee" table in the production schema, you must now use `Prod.Employee` or `Employee` (change the default schema to "Prod") in your SQL statement. To get around this issue, you can create a synonym for the "Test.Employee" table (say "EMP"), and then use it in your SQL statements. Whenever you need to perform operation on the "Prod.Employee" table, modify the definition of the "EMP" synonym to point it to the "Prod.Employee" table. This ensures that you do not have to modify your SQL statements to perform operation on the object in different schemas.

- **Changes in the underlying objects:** The synonyms insulate you from any changes in the name or location of the underlying objects on which you are performing an operation. You can modify the synonym definition to accommodate any changes in the name or location of the underlying objects.

For example, suppose that you are using a table in one of your stored procedures. Now, if the table name changes or the table is moved to some other location then your stored procedure will stop working. To work around this, you can use a synonym for the table in the stored procedure, and update the synonym definition if there is a change in the name or location of the table.

- **Simplified and secure access:** In a distributed environment, you must use the schema name along with the object names to ensure that you are accessing the correct object. Moreover, you must also ensure that the user has required privileges on the target object. To simplify this, you can assign a simple name for an object by creating a synonym that has the full qualified path to the object, and then grant appropriate privileges on the synonym.

Working with Synonyms in the Adapter

The Oracle E-Business adapter exposes the synonyms in Oracle for:

- Tables
- Views
- Stored Procedures
- Functions
- Packages

The synonyms for each of these artifacts are exposed alongside the respective underlying artifact in the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, and Add Adapter Service Reference Plug-in. For example, the **Tables** node under the **Schema-based view** will display all the synonyms for tables along with the database tables in a schema, the **Views** node under the **Schema-based view** will display all the synonyms for views along with the database views in a schema, and

so on.

- For synonyms created on tables and views, the same operations are exposed as for the underlying tables and views respectively. For example, if the underlying tables and views contain LOB columns, the synonyms for those tables and views will also expose the Read_<LOBColName> and Update_<LOBColName> operations for table synonyms and the Read_<LOBColName> operation for the view synonyms.
- For synonyms created on stored procedures, functions, and packages, the synonyms are exposed as operations alongside the respective underlying stored procedures, functions, and packages in a schema.

 **Note**

The Oracle E-Business adapter supports only local synonyms. This implies that only those synonyms are supported by the adapter that target the artifacts on the local server.

Moreover, the message actions for the synonyms are the same as the underlying object except for the artifact name on which the action is performed. For example, the message action for the **Select** operation on a table in the SCOTT schema is:

Tables/Select/SCOTT/[**TABLE_NAME**]. If you are performing a Select operation on a synonym for the same table in the SCOTT schema then the message action will be: Tables/Select/SCOTT/[**SYNONYM_NAME**].

When you invoke an operation on a synonym in the adapter, the adapter calls the synonym in the Oracle database to execute the operation. However, the adapter uses the underlying object name in the synonym definition to fetch the metadata.

Synonyms can be used in normal outbound operations, composite operations, and polling.

 **Note**

You can search for synonyms in Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in just like other objects. However, you cannot search for procedures inside synonym packages from a skip-level node as you can do for the procedures inside packages. For information about searching for operations in the adapter, see [Searching for Operations](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Receiving Database Change Notifications

The Microsoft BizTalk Adapter for Oracle E-Business Suite supports the ODP.NET Database Change Notification feature. Using this feature, the adapter clients can register a SELECT statement as the notification query on the database, and the database sends a notification to the adapter client as and when the result set of the SELECT statement changes. The database change notification is implemented in the adapter using the OracleDependency class. For more information about the Database Change Support feature in ODP.NET and the OracleDependency class, see <http://go.microsoft.com/fwlink/?LinkId=124801>.

The Oracle E-Business adapter exposes an inbound operation, Notification, to support database change notification. However, for the database change notification to work with Oracle E-Business adapter, you must ensure the following:

- Connect to Oracle E-Business Suite with the underlying Oracle database version 10.2 or later. Oracle database versions prior to 10.2 do not support notifications.
- Connect to Oracle E-Business Suite as a user that has the CHANGE NOTIFICATION privilege to create a notification registration. To grant the CHANGE NOTIFICATION privilege to a user, connect to the Oracle database as a user with administrative privileges, and run the following command at the SQL prompt:

```
grant change notification to <user name>
```

- Decide on a TCP port that can be used by ODP.NET to receive database change notifications from Oracle database. Add the TCP port to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkID=196959>. You must provide the same TCP port number for the **NotificationPort** binding property. For more information about the binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

A typical database change notification using the Oracle E-Business adapter involves the following:

1. The adapter clients must specify **Notification** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is Polling.
2. The adapter clients must specify a SQL SELECT statement to register for database change notifications in the **NotificationStatement** binding property. The adapter client gets a notification from Oracle database as and when the result set for the specified SQL statement changes.
3. The adapter clients must specify whether the adapter sends a notification to the adapter clients as soon as the listener is started in the **NotifyOnListenerStart** binding property.
4. The notification is sent to the adapter clients as and when the result set of the SELECT statement specified in the **NotificationStatement** binding property is changed.

⚠ Caution

If there is a network outage between the Oracle database and the adapter client, the notifications will not be sent to the adapter clients for the changes done on the Oracle database during the period of network outage, and thereafter. Therefore, you must use the Polling operation instead of the Notification operation for critical scenarios.

Differences between Notification and Polling

Though notification and polling are both inbound operations, and inform the adapter clients about the data changes in the Oracle database, the following table illustrates some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Notification	Polling
Notification is supported only for Oracle database versions 10.2 and later.	Polling is supported for all the Oracle database versions that are supported by the Oracle E-Business adapter.

<p>The data-change notification is always instantaneous.</p>	<p>You can either configure the polling interval to check the data available for polling at regular intervals or instantaneously as and when the data is available.</p> <p> Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last change notification.</p>
<p>Notification is initiated by the Oracle database. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement. Notification is a feature of the Oracle database.</p>	<p>Polling is initiated by the adapter. The adapter executes a SQL statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.</p>
<p>You can use the notification statement to only read data in an Oracle database.</p>	<p>You can use the polling statement to read or update data in the Oracle database.</p>
<p>Notification informs only about the type of change in the data such as Insert, Update, and Delete.</p>	<p>Polling informs you about the actual data that has changed.</p>

For more information about:

- The binding properties related to the Notification operation, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
- How to use the Notification operation in Oracle E-Business adapter, see [Receiving Database Change Notifications Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Inbound Calls Using Polling

The Microsoft BizTalk Adapter for Oracle E-Business Suite enables client programs to receive messages from Oracle E-Business Suite informing them of changes to data in Oracle E-Business Suite. The Oracle E-Business adapter supports receiving "polling-based" messages wherein the adapter executes a specified SQL statement, stored procedure, function or a procedure within a package, retrieves the data, and provides the result to the client at regular intervals of time.

Note

You can also set the applications context for the Poll operation in Oracle E-Business adapter. It is mandatory to set the applications context for the Poll operation if the operation is performed on an interface table or interface view. For information about applications context, and how to set it, see [Setting Application Context](#).

A typical polling operation using the Oracle E-Business adapter involves the following:

1. The adapter clients must specify **Polling** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is **Polling**.
2. The adapter clients must specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine whether there is data available for polling. If the first column of the first row of the first result set returned on execution of this statement contains a positive integer value, there is data available for polling.
3. The adapter clients must specify a polling interval for the **PollingInterval** binding property to define the interval in seconds at which the statement specified in the **PolledDataAvailableStatement** binding property is executed. At the end of every polling interval, the polled data available statement is executed, and the result set is returned.
4. The adapter clients must specify a SELECT statement or a stored procedure for the **PollingInput** binding property. If you want to poll a table or view, you must specify a SELECT statement for this binding property. If you want to poll using a stored procedure, you must specify the entire request message for this binding property.

The statement in the **PollingInput** binding property is executed only if there is data available for polling, which is determined by the **PolledDataAvailableStatement** binding property in step 2.

5. The adapter clients must specify an action for the polling operation in the **PollingAction** binding property. The polling action for a specific operation is determined from the metadata generated for the operation using the Consume Adapter Service Add-in.
6. The adapter clients can use the **PollWhileDataFound** binding property to ignore the polling interval, and continuously poll data, as and when available.

Important

If you set the value of the **PollWhileDataFound** binding property to True, the adapter clients continuously poll data from Oracle and in the process open and close connections to the Oracle database in a loop. As the rate at which connections are opened by ODP.NET is greater than the connections being closed, the connections get exhausted after some time, and an exception is thrown. As a work around, make sure that the value of the **UseOracleConnectionPool** is set to True, and an appropriate value is mentioned in the **IncrPoolSize** binding property to control the number of connections that can be opened by the adapter clients.

7. The adapter clients can specify a post-poll statement, an Oracle PL/SQL block, for the **PostPollStatement** binding property. The statement specified in this binding property is executed after the statement specified in the **PollingInput** binding property is executed.

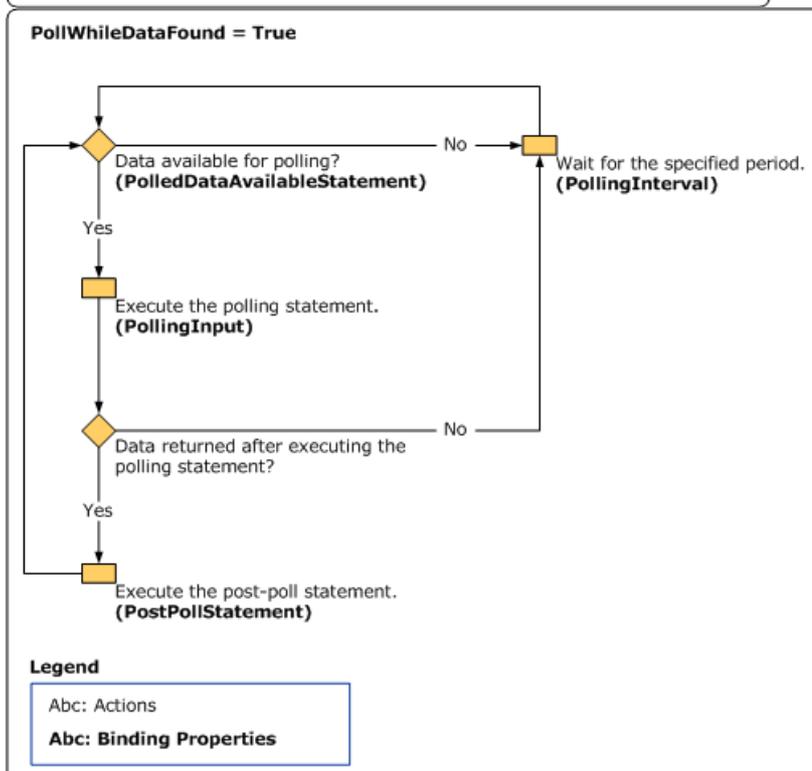
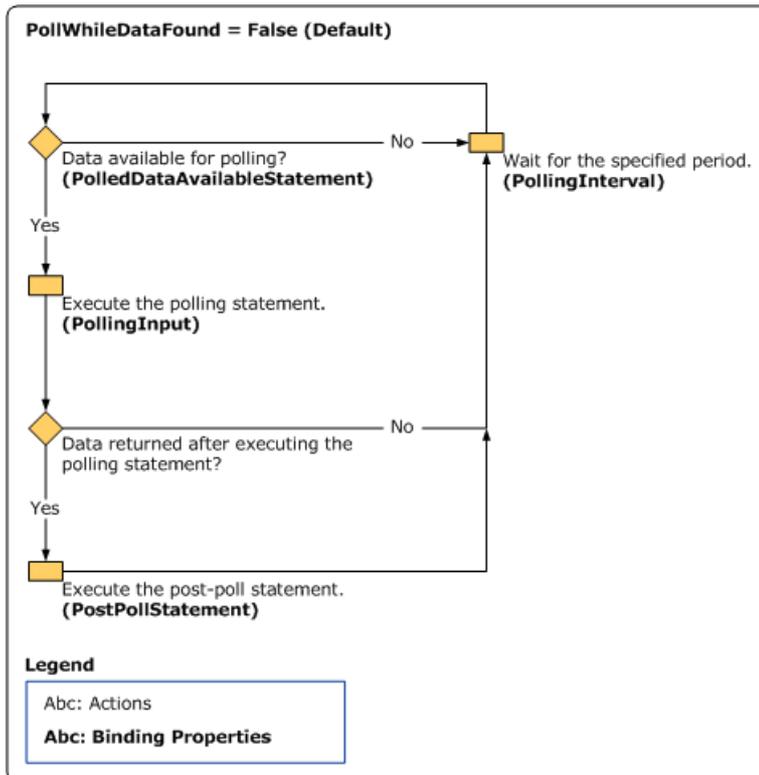
Note

The adapter executes the statement specified in the **PollingInput** and **PostPollStatement** binding properties in a transaction. For more information about transactions in the Oracle E-Business adapter, see [How does the Adapter Handle Transactions?](#).

The adapter suppresses any empty polling responses coming from Oracle E-Business Suite.

The illustration below provides information about the polling workflow in Oracle E-Business adapter. Two scenarios for the polling workflow are illustrated:

1. When the value of the **PollWhileDataFound** is set to "False" (default setting).
2. When the value of the **PollWhileDataFound** is set to "True."



Differences between Polling and Notification

Though polling and notification are both inbound operations, and inform the adapter clients about the data changes in the Oracle database, the following table lists some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Polling	Notification
Polling is supported for all the Oracle database versions that are supported by the Oracle E-Business adapter.	Notification is only supported for Oracle database versions 10.2 and later.
You can either configure the polling interval to check the data available for polling at regular intervals or instantaneously as and when the data is available.  Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last change notification.	The data-change notification is always instantaneous.
Polling is initiated by the adapter. The adapter executes a SQL statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.	Notification is initiated by the Oracle database. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement. Notification is a feature of the Oracle database.
You can use the polling statement to read or update data in the Oracle database.	You can use the notification statement to only read data in an Oracle database.
Polling informs you about the actual data that has changed.	Notification informs only about the type of change in the data such as Insert, Update, and Delete.

For more information about:

- The binding properties related to polling, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
- Receiving polling-based messages using BizTalk Server, see [Polling Oracle E-Business Suite Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes the following outbound operations at the root level:

- **ExecuteNonQuery:** Use this operation to execute any arbitrary SQL statements or PL/SQL blocks in Oracle E-Business Suite if you want to return multiple result sets. The input parameters of this function include a string parameter (the entire PL/SQL block to be executed) and an array of strings (OutRefCursorNames). Each string value specified in OutRefCursorNames is assumed to be the parameter name of an output REF CURSOR with the PL/SQL block returning REF CURSORS with the same names. This function also takes an OUT parameter (OutRefCursors), which is an array of DataSets. For information about DataSet, consult the Oracle documentation at <http://go.microsoft.com/fwlink/?LinkId=124538>. The return value of this operation is of integer data type, and indicates the number of affected rows.
- **ExecuteReader:** Use this operation to execute any arbitrary SQL statements or PL/SQL blocks in Oracle E-Business Suite if you want the result set to be returned as DataSet. This operation takes a string parameter as input, and returns a DataSet.
- **ExecuteScalar:** Use this operation to execute any arbitrary SQL statements or PL/SQL blocks in Oracle E-Business Suite if you want only one value to be returned. If the return value is a result set, only the value in the first column of the first row is returned in a XML string format.

Note

- The ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations are not supported for the User Defined Types (UDTs).
- You can also set the applications context for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations in Oracle E-Business adapter. It is mandatory to set the applications context for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations if any of the operation is targeted on an artifact in Oracle E-Business Suite (interface table, interface view, concurrent programs or request sets). For information about applications context, and how to set it, see [Setting Application Context](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Composite Operations

The Oracle E-Business adapter enables adapter clients to perform composite operations that can include any number of the following operations, and in any order:

- Select, Insert, Update, and Delete operations on an interface table and database table.
- Select operation on an interface view and database view.
- Stored procedures, functions, and procedures within packages that are surfaced as operations in the adapter.

The operations in a composite operation can target tables and views in the same database or different databases. However, data cannot be shared or reused across different operations in a composite operation. For example, in a composite operation, the result set of a Select operation cannot be used as the input parameter for a stored procedure.

Each operation in a composite operation is performed using a separate connection. The Oracle E-Business adapter consumes as many connections from the ODP.NET connection pool as the number of operations in a composite operation, and then releases the connections as the operations get executed. However, if an operation in the composite operation returns a result set, the connection is released only after the message is consumed.

Important

If you experience time-out issues while executing a composite operation then it could be because the number of connections is less than the number of operations in a composite operation involving:

- Stored procedures containing BFILE, BLOB, CLOB, NCLOB, and REF CURSOR as OUT or IN OUT parameters.
- Select operation.

To resolve this issue, you must ensure that if there are "n" number of such operations in a composite operation, the value specified for the **MinPoolSize** binding property is "n+1" or greater. For more information about the **MinPoolSize** binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

For information about:

- How to perform composite operations in Oracle E-Business adapter using BizTalk Server, see [Performing Composite Operations on Oracle Database by Using BizTalk Server](#).
- Message schemas for the composite operation, see [Message Schemas for the Composite Operation](#).

Note

You can also set the applications context for composite operations in the Oracle E-Business adapter. It is mandatory to set the applications context for the composite operations if any of the operations in a composite operation is performed on an interface table or interface view. For information about applications context, and how to set it, see [Setting Application Context](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Oracle User-Defined Types

The Microsoft BizTalk Adapter for Oracle E-Business Suite supports performing operations on artifacts in Oracle E-Business Suite and the underlying database that contain Oracle User-Defined Types (UDTs). The UDTs can be present in the following artifacts:

- Interface tables and interface views containing UDT columns.
- Database tables and views containing UDT columns.
- Packages, stored procedures, and functions containing UDT parameters.

What is an Oracle UDT?

Oracle UDTs help in representing complex entities as a "single" object that can be shared among the applications. For example, it is possible to model real-world entities such as "Customers" or "Sales Orders" as objects in the Oracle database. Oracle UDTs are defined in the Oracle database, and they are of the following two types:

- Object types. For example, Oracle Object.
- Collection types. For example, nested table types or VARRAY.

The name of the Oracle UDT is case sensitive, and must be specified in the following way: [SCHEMA_NAME].[UDT_NAME].

How Does the Adapter Support Oracle UDT?

ODP.NET supports UDTs by representing Oracle UDTs defined in the Oracle database as .NET types (custom types). Custom types define the mapping between the Oracle UDT attributes or elements to the .NET members. Custom types can be .NET classes or structures, and can represent either Oracle Objects or Oracle Collections. Owing to the fact that the Oracle E-Business adapter uses ODP.NET to connect to the Oracle database, it inherits support for Oracle UDTs.

The Oracle E-Business adapter uses the ODP.NET to specify a custom type mapping to map a .NET custom type to an Oracle UDT in the database. To specify a custom type mapping, the Oracle E-Business adapter uses a custom type factory. Therefore, in order to use an Oracle UDT, an assembly (.dll file) is required that defines the custom type factory. The Oracle E-Business adapter enables you to generate an assembly for the custom type factory while generating the metadata for an artifact/operation that contains an Oracle UDT.

Note

The adapter generates the assembly for the Oracle UDTs based on the classes used by the ODP.NET to support Oracle UDTs. For detailed information about how Oracle UDTs are supported in ODP.NET, see <http://go.microsoft.com/fwlink/?LinkId=140697>.

To generate the assembly file for using the Oracle UDTs at design time and then use it later at the run time, the Oracle E-Business adapter exposes the following binding properties:

- **GeneratedUserTypesAssemblyFilePath** (design time)
- **GeneratedUserTypesAssemblyKeyFilePath** (design time)
- **UserAssembliesLoadPath** (run time)

For information about these binding properties, see

[Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Performing Operations On Artifacts Containing Oracle UDTs

To perform operations on artifacts containing UDTs using the Oracle E-Business adapter, you must do the following during design time and run time.

Design Time

You must perform these steps while generation schema for the operation in Visual Studio.

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in. For information about doing so, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).
2. While connecting, in the **Binding Properties** tab of the **Configure Adapter** dialog box, specify appropriate values for the **GeneratedUserTypesAssemblyFilePath** and **GeneratedUserTypesAssemblyKeyFilePath** binding properties. For information about these binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
3. When you are connected to Oracle E-Business Suite in Visual Studio, browse to the required artifact that contains an Oracle UDT. For information about browsing artifacts, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).
4. Select the required artifact, and then click **OK**. The Oracle E-Business adapter generates the metadata for the selected operation along with the assembly (.dll file) for the Oracle UDT in the selected artifact. The assembly is created at the location that you specified in the **GeneratedUserTypesAssemblyFilePath** binding property.
5. Proceed with the rest of the steps for building and deploying your project.

Run Time

You must perform these steps in the adapter clients to perform operations on the Oracle UDTs.

In BizTalk Server

- Manually add the Oracle UDT assembly created in step 4 in "Design Time" to the Global Assembly Cache (GAC) on your computer. Alternatively, you can manually copy the Oracle UDT assembly under the BizTalk Server installation location. For BizTalk Server 2010, typically this is <installation drive>:\Program Files\Microsoft BizTalk Server 2010.
- While configuring the BizTalk Server WCF-Custom or WCF-OracleEBS port, in the **Binding** tab, specify the location of the Oracle UDT assembly for the **UserAssembliesLoadPath** binding property. For information about this binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

In Visual Studio

- Manually add the Oracle UDT assembly created in step 4 in "Design Time" to the Global Assembly Cache (GAC) on your computer. Alternatively, you can manually copy the Oracle UDT assembly to the same location as the project executable file, which typically is under the project's \bin\Debug folder.
- Specify the location of the Oracle UDT assembly for the **UserAssembliesLoadPath** binding property. For information about this binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

How does the Adapter Handle Transactions?

Microsoft BizTalk Adapter for Oracle E-Business Suite does not initiate a transaction while performing an operation in Oracle E-Business Suite. Instead, the adapter performs the operations using the transaction context provided by the adapter clients. In order to perform operations in a transaction using the Oracle E-Business adapter, you must:

- Enable transactions in the adapter clients. For example, to enable transactions in BizTalk Server 2010, you must select the **Use Transaction** check box in the **Transactions** area of the **Messages** tab for a WCF-Custom or WCF-OracleEBS port.
- Set the value of the **UseAmbientTransaction** binding property to **True** in the adapter. For more information about the binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

◆ Important

To use the adapter to perform transactions in Oracle E-Business Suite, you must have installed the **Oracle Services For Microsoft Transaction Server** component, while installing the Oracle client, on the computer running the adapter client.

Transactions in the Outbound Operations

The Oracle E-Business adapter performs an outbound operation in a single transaction. For composite operations, all the operations are performed in a single transaction but using different ODP.NET connections. For more information about the outbound operations surfaced by the Oracle E-Business adapter, see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#)

Transactions in the Inbound Operations

The Oracle E-Business adapter exposes the following two inbound operations:

- **Polling:** The polling statement and the post-poll statement (if specified) are executed in a transaction, whereas, the polled data available statement is executed in a different transaction. Similarly, the polling statement and the post-poll statement are executed using the same ODP.NET connection, whereas, the polled data available statement is executed using a different ODP.NET connection.
- **Notification:** The notification operation is performed in a transaction using a single ODP.NET connection.

For more information about the inbound operations surfaced by the Oracle E-Business adapter, see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#)

See Also

Other Resources

[Overview of BizTalk Adapter for Oracle E-Business Suite](#)

Other Features Supported by the Adapter

In addition to the features discussed throughout the topics of [Overview of BizTalk Adapter for Oracle E-Business Suite](#), the Microsoft BizTalk Adapter for Oracle E-Business Suite also provides the following features that are useful for adapter clients:

- **Support for configuring adapters using binding properties.** Adapter clients can configure the Oracle E-Business adapter by specifying certain binding properties. For example, clients can configure the adapter to use the ODP.NET connection pool by setting the **UseOracleConnectionPool** binding property. For more information, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
- **Support for null values for operation parameters.** Adapter clients can provide null values for operation parameters using the "nil" attribute in the input XML.
- **Support for dynamic ports in BizTalk.** Through the BizTalk WCF-Custom adapter, the Oracle E-Business adapter supports a dynamic port that enables dynamic routing of messages from BizTalk Server based on the message context properties. For more information, see [Configuring Dynamic Ports](#).

See Also

Other Resources

[Overview of BizTalk Adapter for Oracle E-Business Suite](#)

Key Features in BizTalk Adapter for Oracle E-Business Suite

This section lists the key features in Microsoft BizTalk Adapter for Oracle E-Business Suite.

Technology-Related Features

Feature	Comment
Use of Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK (WCF LOB Adapter SDK)	The Oracle E-Business adapter is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK (WCF LOB Adapter SDK). In turn, the WCF LOB Adapter SDK is built on top of WCF. The adapters are exposed as WCF channels to adapter clients. This enables connectivity, metadata exchange, and business data exchange with external systems.
Support for the WCF channel model and the WCF service models	In the WCF <i>channel</i> model, adapter clients can consume the Oracle E-Business adapter by directly sending and receiving XML messages. In the WCF <i>service</i> model, adapter clients can generate a .NET proxy class from the Web Services Description Language (WSDL) obtained using Oracle E-Business adapter.
Use of ODP.NET	The Oracle E-Business adapter uses the Oracle Data Provider for .NET (ODP.NET) to interface with Oracle E-Business Suite.
Two ways of connecting to Oracle E-Business Suite	The adapter clients can connect to Oracle E-Business Suite by either using the net service name in the tnsnames.ora file or by directly specifying the connection parameters, and thus eliminating the need to use a net service name or the tnsnames.ora file. Not requiring the tnsnames.ora file to connect to Oracle E-Business Suite saves you from the hassle of manually updating the connection parameters (net service name) in the tnsnames.ora file on every client computer when you add or update Oracle servers in your environment. For more information, see Establishing a Connection to the Oracle E-Business Suite .
Support for Windows Authentication	The adapter clients can use Windows Authentication to connect to Oracle E-Business Suite. Windows Authentication enables you to determine the user's identity based on the Windows logon credentials, and thus helps you to leverage the built-in security of the Windows environment. For more information about Windows Authentication in the Oracle E-Business adapter, see Connecting to Oracle E-Business Suite Using Windows Authentication .
Support for using the Oracle E-Business adapter with Microsoft Office SharePoint Server (MOSS)	You can use the adapters to present data from the Oracle E-Business Suite onto a MOSS portal. For more information, see Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server .

Metadata-Related Features

Feature	Comment
Batch retrieval of metadata	Adapter clients can browse and search metadata for all the artifacts exposed by Oracle E-Business Suite and the underlying database. The artifacts are displayed based on the connected user credentials, and are: <ul style="list-style-type: none"> Grouped by each application for the Oracle E-Business Suite artifacts. Grouped by each artifact in Oracle E-Business Suite and the underlying database. Grouped by each schema for the underlying database artifacts.

Performance-Related Features

Feature	Comment
---------	---------

Connection pooling	The Oracle E-Business adapter enables clients to use the ODP.NET connection pooling by configuring the UseOracleConnectionPool binding property. For more information about this binding property, see Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties .
Cache management	The Oracle E-Business adapter enables clients to use the ODP.NET cache management by configuring the StatementCachePurge and StatementCacheSize binding properties. For more information about this binding property, see Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties .
Support for performance counters	The Oracle E-Business adapter supports WCF-based performance counters for use by adapter clients. For more information about performance counters, see Using Performance Counters .

Operations-Related Features

Feature	Comment
Support for operations on interface tables and interface views	Adapter clients can perform basic Insert, Update, Delete, and Select operations on Oracle E-Business Suite interface tables. Only select operation can be performed on the Oracle E-Business Suite interface views.
Support for executing PL/SQL APIs	Adapter clients can execute PL/SQL APIs in Oracle E-Business Suite. PL/SQL APIs contain stored procedures and functions within a package.
Support for executing concurrent programs	Adapter clients can execute concurrent programs in Oracle E-Business Suite.
Support for executing request sets	Adapter clients can execute request sets in Oracle E-Business Suite. A request set is a set of concurrent programs, and can be used to run several concurrent programs with predefined parameters. Request sets are executed in stages. For more information about request sets, see http://go.microsoft.com/fwlink/?LinkId=129539 .
Support for specifying inline values in the Insert operation	You can use the InlineValue attribute in the Insert operation to insert computed values into interface tables and database tables in Oracle. This is an optional attribute and is available for all simple data records in an Insert operation. If you specify a value for this attribute, it overrides the specified value of a record. For more information about the InlineValue attribute, see Operations on Interface Tables and Interface Views .
Support for inbound calls using polling	The Oracle E-Business adapter supports receiving "polling-based" data-changed messages wherein the adapter executes a specified SQL statement, stored procedure, function, or procedure within a package, retrieves the data, and provides the result to the client.
Support for invoking functions and procedures	Adapter clients can invoke functions and procedures in the underlying Oracle database. The Oracle E-Business adapter surfaces functions and procedures as operations that can be performed on the underlying Oracle database.
Support for composite operations	The Oracle E-Business adapter enables adapter clients to perform composite operations in Oracle E-Business Suite. A composite operation can include any number of the following operations, and in any order: <ul style="list-style-type: none"> • Operations on interface tables, interface views, tables, and views. • Stored procedures, functions, and procedures within packages that are surfaced as operations in the adapter.

Support for executing arbitrary SQL statements and PL/SQL blocks	The Oracle E-Business adapter enables adapter clients to execute arbitrary SQL statements or PL/SQL blocks using the ExecuteReader, ExecuteScalar, and ExecuteNonQuery operations. For more information about these operations, see Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations .
Support for User-Defined Types (UDTs), boolean parameters, and PL/SQL table types	Adapter clients can support operations in artifacts that contain UDTs, and operations in stored procedures and functions that contain boolean parameters and PL/SQL table type. For information about UDT support, see Support for Oracle User-Defined Types .
Support for input and output REF CURSORS	Adapter clients can operate with input and output REF CURSORS on procedures and functions.
Support for PL/SQL RECORD data types	Adapter clients can operate with RECORD and nested RECORD types in stored procedures and functions.
Support for overloaded functions and procedures in PL/SQL API	Adapter clients can invoke overloaded functions and procedures in PL/SQL APIs. However, the Oracle E-Business adapter does not support the use of overloaded procedures using strongly-typed and weakly-typed REF CURSORS. Internally, the adapter treats both the strongly-typed and weakly-typed REF CURSORS as just REF CURSORS.
Support for database change notifications	Adapter clients can receive database change notifications from Oracle database based on a triggering SELECT statement. The notification is sent by Oracle database to the adapter clients as and when the result set for the SELECT statement changes. For more information about database change notifications, see Receiving Database Change Notifications .
Support for synonyms	Adapter clients can perform operations on synonyms created for tables, views, stored procedures, functions and packages. For more information about synonyms, and how you can use the Oracle E-Business adapter to perform operations on synonyms, see Operations on Synonyms .

Other Features

Feature	Comment
Multi-Language Support (MLS)	<p>Adapter clients can use the Oracle E-Business adapter to perform operations in Oracle E-Business Suite and the underlying database having multiple language packs installed. This implies that the adapter clients can perform operations on the data in multiple languages. For example, you can view localized (non-English) data in an interface view, or insert localized data in a table. The Oracle E-Business adapter supports as many languages as those supported by the underlying Oracle E-Business Suite installation.</p> <p>The Oracle E-Business adapter exposes the MlsSettings binding property to configure the multiple language support feature. For more information about the binding property, see Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties.</p>

See Also

Other Resources

[Understanding BizTalk Adapter for Oracle E-Business Suite](#)

Limitations of BizTalk Adapter for Oracle E-Business Suite

General Limitations

The following are known limitations for Microsoft BizTalk Adapter for Oracle E-Business Suite:

- The Oracle E-Business adapter does not support XML Gateway, Advanced Queuing, and Business Events.

However, you can get around the Business Events limitation in the following way:

1. In Oracle Business Events System, create a subscription to invoke a custom PL/SQL procedure when a business event occurs.
2. Write a custom PL/SQL procedure that receives the business event.
3. Use the custom PL/SQL procedure to store the resultant data (event and event payload) in a table.
4. Use the Oracle E-Business adapter to poll or receive notifications from the table.

- The Oracle E-Business adapter does not support XML Types.
- The Oracle E-Business adapter does not enable clients to set the value of the first element in a VARRAY to NULL.
- The Oracle E-Business adapter does not support records that contain fields of type PL/SQL tables of RECORD type.
- The Oracle E-Business adapter does not support User-Defined Types (UDTs) that have circular references.
- The Oracle E-Business adapter does not support the BFILE data type inside complex types (such as RECORD type, TABLE type, UDT, and VARRAY).
- The Oracle E-Business adapter supports UDT nesting only up to two levels.
- Except for PL/SQL tables, the Oracle E-Business adapter does not support UDTs that are defined inside a package.
- When using the adapters with BizTalk Server, if the credentials on the WCF-custom send port are incorrect, the request messages are not processed. After you specify the correct credentials, the message is sent to Oracle E-Business Suite and a response is received. However, the response message is not available to the out port. In such scenarios, you may need to restart the host instance.

Limitations Due to ODP.NET

The following are known limitations for Oracle E-Business adapter due to the limitation of ODP.NET:

- The Oracle E-Business adapter does not support PL/SQL tables that are not indexed by a numeric field.
- The Oracle E-Business adapter does not support associative arrays that do not contain any element.
- The Oracle E-Business adapter does not support UDTs that contain the TimeStamp data type with local time zone attributes (TimeStampLTZ).
- The Oracle E-Business adapter does not support UDTs that contain a "." (period) in their names.
- The Oracle E-Business adapter does not support UDTs that contain BLOB, CLOB, and NCLOB data types as an IN OUT parameter.

- The Oracle E-Business adapter does not support Varray of Varray of the following simple types: BFILE, IntervalDS, IntervalYM, TimeStampLTZ, and TimeStampTZ.
- Due to the limitation of associative arrays, PL/SQL tables or PL/SQL tables of records that contain any of the following data types are not supported in the Oracle E-Business adapter:
 - BFILE
 - BLOB
 - CLOB
 - IntervalDS
 - IntervalYM
 - Long
 - NCLOB
 - RowID
 - TimeStamp
 - TimeStampLTZ
 - TimeStampTZ

See Also

Other Resources

[Understanding BizTalk Adapter for Oracle E-Business Suite](#)

Oracle E-Business Adapter Tutorial

The Microsoft BizTalk Adapter for Oracle E-Business Suite tutorial contains a step-by-step demonstration of how to present data from Oracle E-Business Suite system on a Microsoft SharePoint portal. It also demonstrated how you can configure a search application on the SharePoint portal that can search for an entity in a sample artifact in Oracle E-Business Suite.

In addition to the tutorial, the topics in the [Developing BizTalk Applications](#) section provide detailed instructions on how to use the Oracle E-Business adapter with BizTalk Server.

In This Section

[Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#)

See Also

Other Resources

[Oracle E-Business Adapter Tutorial](#)

Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site

This tutorial provides detailed instructions on using the Oracle E-Business adapter with Microsoft Office SharePoint Server to present data from Oracle E-Business Suite on a SharePoint portal. It also demonstrates how to configure a search application in the Microsoft Office SharePoint Server that allows you to do a full-text search from a SharePoint portal on the configured Oracle E-Business Suite artifact.

To demonstrate how to do so, consider a sample interface table in Oracle E-Business Suite that stores information about employees. In this tutorial, an application is created in Microsoft Office SharePoint Server that retrieve a list of customers from Oracle E-Business Suite based on a search string using:

- A Business Data List Web Part in Microsoft Office SharePoint Server
- The search feature in Microsoft Office SharePoint Server

Note

Before proceeding with the tutorial, make sure you have installed all the prerequisites for using the Oracle E-Business adapter with Microsoft Office SharePoint Server. For information about the prerequisites, see the BizTalk Adapter Pack installation guide, typically installed at C:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

In This Section

- [Step 1: Use the Oracle E-Business Adapter to Create and Publish a WCF Service](#)
- [Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts](#)
- [Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite](#)
- [Step 4: Test Your SharePoint Application](#)

Step 1: Use the Oracle E-Business Adapter to Create and Publish a WCF Service



Time to complete: 15 minutes

Objective: You can use the WCF LOB Adapter Service Development Wizard to generate a WCF service from the Oracle E-Business Suite artifacts that can be hosted in a hosting environment such as Internet Information Services (IIS) or Windows Process Activation Service (WAS). This topic demonstrates how to use the wizard to generate a WCF service file.

Prerequisites

Before running the wizard, install the following:

- Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK either with the **Complete** option or the **Custom** option (and choosing **Tools** within this option). This installs the Visual Studio template for the Adapter Service Development Wizard.
- Microsoft BizTalk Adapter for Oracle E-Business Suite from the BizTalk Adapter Pack.

For more information about these prerequisites, see the BizTalk Adapter Pack installation guide. The installation guide is typically installed at <installation drive>\Program Files\Microsoft BizTalk Adapter Pack\Documents.

Note

You must also run the create_apps_artifacts.sql script provided with the Microsoft Office SharePoint Server sample to create the **MS_SAMPLE_EMPLOYEE** interface table in the **Application Object Library** application. This interface table is used in this tutorial.

In this topic

We will perform the following tasks in this topic:

1. [Create a WCF Service for the Select Operation on the MS_SAMPLE_EMPLOYEE Interface Table](#)
- 2.
3. [Modify the .cs file](#)
4. [Publish the WCF Service](#)

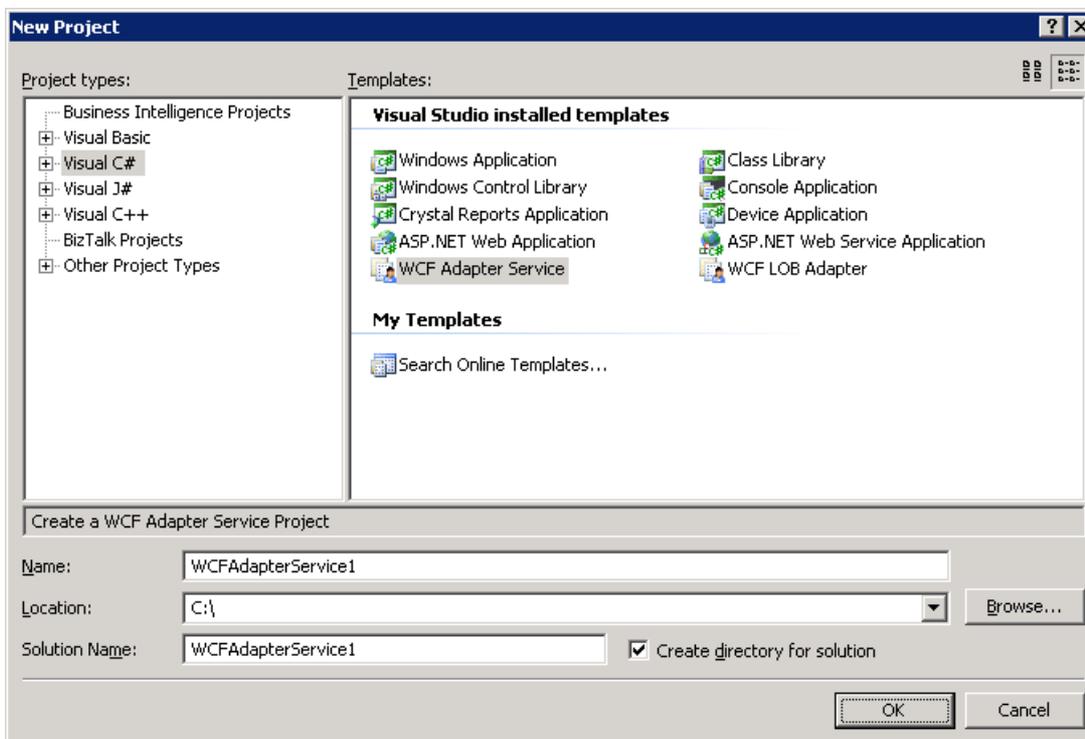
Create a WCF Service for the Operation on Oracle E-Business Artifact

This section provides the steps to create a WCF service for the Select operation on the MS_SAMPLE_EMPLOYEE Interface Table.

To Create a WCF Service for the Select Operation on the MS_SAMPLE_EMPLOYEE Interface Table

1. Start Visual Studio, and then create a project.
2. In the **New Project** dialog box, from the **Project types** pane, select **Visual C#**. From the **Templates** pane, select **WCF Adapter Service**.

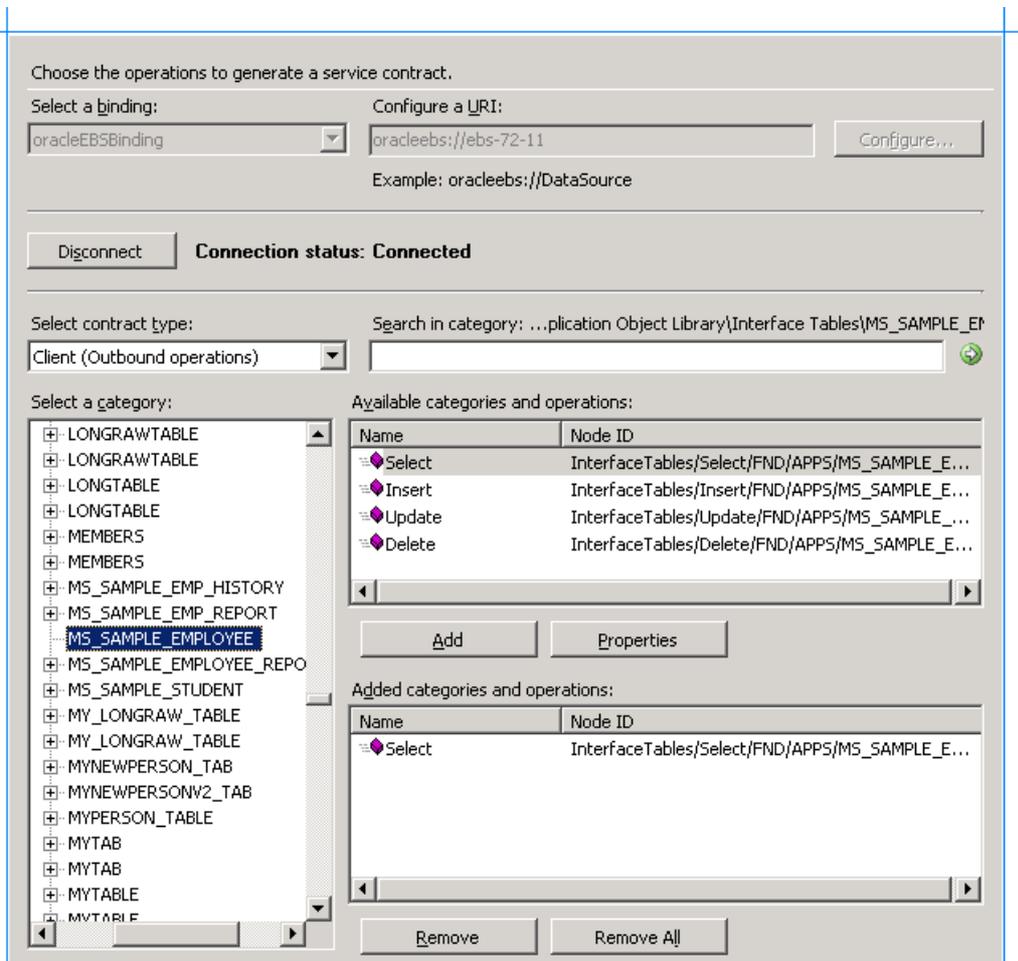
Alternatively, from the **Project types** pane, expand **Visual C#**, and then select **Web**. From the **Templates** pane, select **WCF Adapter Service**.



Note
 If you installed Visual Studio 2010 with the Web Development component, the **WCF Adapter Service** template is also available from the **New Web Site** option (**File > New > Web Site**).
 However, the Oracle E-Business adapter only supports Web sites that are created on the file system. Therefore, while creating a Web site in the New Web Site dialog box, you must click File System in the Location list.

3. Specify a name and location for the solution, and then click **OK**. The WCF Adapter Service Development Wizard starts.
4. On the Welcome page, click **Next**.
5. On the Choose Operations page, specify a connection string to connect to Oracle E-Business Suite. To do so:
 - a. In the **Select a binding** list, click **oracleEBSBinding**, and then click **Configure**.
 - b. In the **Configure Adapter** dialog box, click the **Binding Properties** tab.
 - i. Under the **General** category, for the **ClientCredentialType** binding property, select **EBusiness**.
 - ii. Under the **OracleEBS** category, specify appropriate values for the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. In this case, you need to provide database credentials for the **OracleUserName** and **OraclePassword** binding properties.
 - iii. Under the **Metadata** category, for the **EnableSafeTyping** binding property, select **True**. If you are retrieving values for the date column, we recommend that you set the **EnableSafeTyping** binding property to **True** when generating the metadata.
 - c. Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).
 - d. Click the **Security** tab, and in the **Client credential type** list, select **Username**. Specify a valid Oracle E-Business Suite user name and password to connect to Oracle E-Business Suite.
 - e. Click **OK** to close the Configure Adapter dialog box, and then click **Connect**. After Visual Studio successfully establishes a connection with Oracle E-Business Suite, the connection status is shown as **Connected**. You can also see the Oracle E-Business Suite metadata being displayed on the Choose Operations page.

6. On the Choose Operations page, in the **Select contract type** list, click **Client (Outbound operations)**.
7. In the **Select a category** box, browse to the MS_SAMPLE_EMPLOYEE interface table in Application Object library application. For information about browsing to an artifact in the adapter, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).
8. In the **Available categories and operations** box, select the **Select** operation, and then click **Add**. The Select operation is added to the **Added categories and operations** box.



Note

You can add more than one operation for each artifact. You can also add operations for different Oracle E-Business Suite artifacts. For example, you can add one operation for an interface table and another for a concurrent program. In addition, you can search for specific operations by specifying wildcard characters in search expressions. For more information about the supported special characters and the node levels at which you can search for the operations, see [Searching for Operations](#).

9. On the Choose Operations page, click **Next**.
10. On the Configure Service and Endpoint Behaviors page, specify values to configure the service and endpoint behavior.
 - a. In the **Service Behavior Configuration** box, specify values for the following:

For the property	Specify the value
EnableMetadataExchange	Set this to True to create a metadata exchange endpoint. By setting this to True , you make the service metadata available using standardized protocols, such as WS-Metadata Exchange (MEX) and HTTP/GET requests. Default is False .

IncludeExceptionDetailsinFault	Set this to True to include managed exception information in detail of SOAP faults returned to the client for debugging purposes. Default is False .
Name	Name for the service behavior configuration. For this tutorial, type customServiceBehavior .
UseServiceCertificate	Specifies whether you want to use the message level security mode of WCF. Default is True . For this tutorial, you must set this to False .
Note	
Because we are not using service certificates for this tutorial, you do not need to provide values for the FindValue , StoreLocation , StoreName , and X509FindType properties. For more information about the certificates and the associated properties, see "X509ClientCertificateCredentialsElement Properties" at http://go.microsoft.com/fwlink/?LinkId=103771 .	

b. In the **Endpoint Behavior Configuration** box, specify values for the following:

For the property	Specify the value
Authentication Type	For Microsoft Office SharePoint Server to consume the WCF service, you should set this as HTTPUserNamePassword . This enables clients to specify user name and password as part of the HTTP header.
Name	Specify a name for the endpoint behavior configuration. For this tutorial, type customEndpointBehavior .
UserNameHeader	Name for the user name header. For this example, specify MyUserHeader . For more information about HTTP headers, see "Support for Custom HTTP and SOAP Headers" at http://go.microsoft.com/fwlink/?LinkId=106692 . Note You must specify a value for this property if the Authentication Type is set to HTTPUserNamePassword . If Authentication Type is set to Auto , this property is optional.
PasswordHeader	Name for the password header. For this example, specify MyPassHeader . For more information about HTTP headers, see "Support for Custom HTTP and SOAP Headers" at http://go.microsoft.com/fwlink/?LinkId=106692 . Note You must specify a value for this property if the Authentication Type is set to HTTPUserNamePassword . If Authentication Type is set to Auto , this property is optional.

The following figure shows the Configure Service and Endpoint Behaviors page with the specified values.

Configure service and endpoint behaviors.

These service and endpoint behavior configurations will be applied for all the generated endpoints.

Service Behavior Configuration

General	
EnableMetadataExchange	True
IncludeExceptionDetailsInFault	True
Name	customServiceBehavior
UseServiceCertificate	False
Service Certificate	
FindValue	
StoreLocation	LocalMachine
StoreName	My
X509FindType	FindBySubjectDistinguishedName

UseServiceCertificate
If this is set to True, you must provide certificate details.

Endpoint Behavior Configuration

General	
Name	customEndpointBehavior
AuthenticationType	HTTPUsernamePassword
HTTP Custom Headers	
UsernameHeader	MyUserHeader
PasswordHeader	MyPasswordHeader

PasswordHeader
Password header

- On the Configure Service and Endpoint Behaviors page, click **Next**.
- On the Configure Service Endpoint Binding and Address page, the **Select a contract to configure** displays the artifact (MS_SAMPLE_EMPLOYEE) that you configured. The **Operations under the selected contract** box displays the **Select** operation you selected for the artifact on the Choose Operations page.
- In the **Configure the address and binding for the contract** box, specify values for the following:

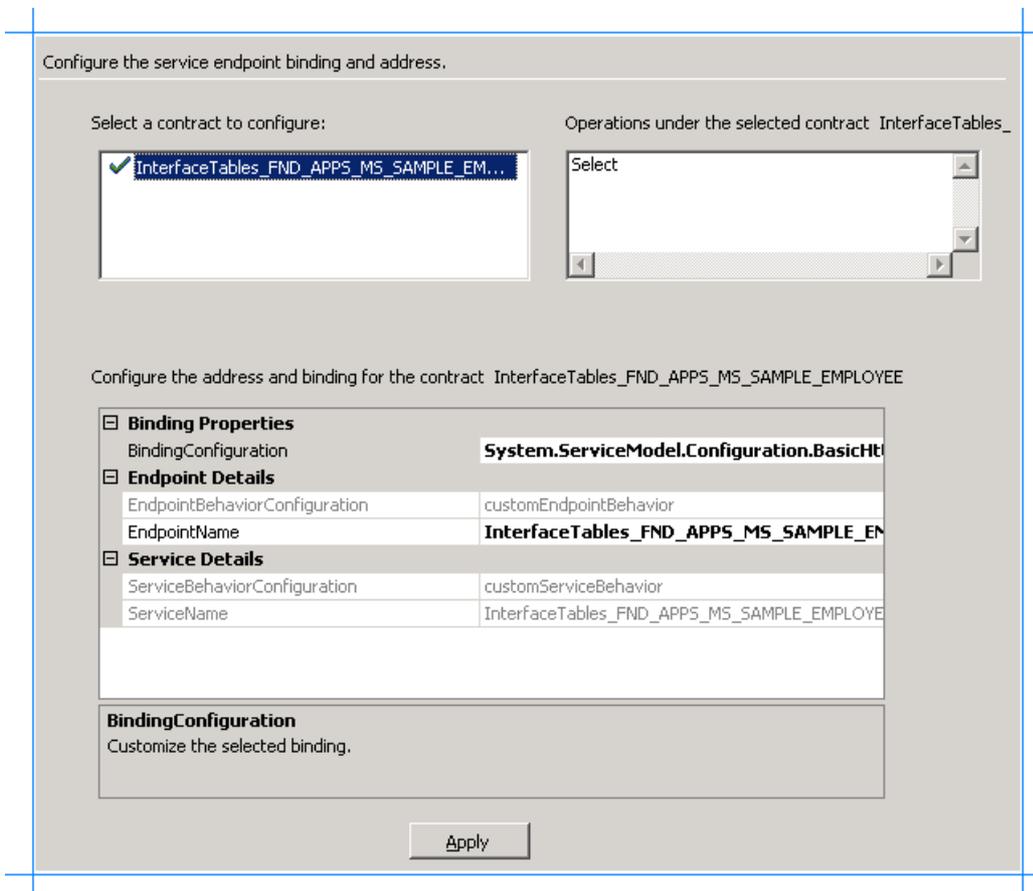
For the property	Specify the value
Binding Configuration	<p>The wizard only supports basic HTTP binding. So, the binding configuration field is automatically populated to <i>System.ServiceModel.Configuration.BasicHttpBindingElement</i>.</p> <p>Click the ellipsis button (...) to change the properties for HTTP binding. To use a secure communication channel, you must always set the Mode property to Transport. The wizard sets the default value for the Mode property as Transport.</p> <p>For more information about the other bindings exposed, see "BasicHttpBindingElement Members" at http://go.microsoft.com/fwlink/?LinkId=103773.</p>
Endpoint Name	Specify an endpoint name for the contract.

The other fields on this page are automatically populated based on the values you specified in the earlier pages.

Click **Apply**.

Note
If you do not specify any values on this page, the default values are accepted for all the contracts.

The following figure shows the Configure Service Endpoint Binding and Address page with the specified values.



14. On the Configure Service Endpoint Binding and Address page, click **Next**. The Summary page lists a tree structure of the Oracle E-Business Suite artifact and the operation selected for the artifact.
15. Review the summary, and then click **Finish**.
16. The wizard creates a WCF service and adds the following files to the Visual Studio project:
 - a. .svc file. This is the WCF service file. The wizard generates one file for each contract.
 - b. Web.config file.
 - c. Service code (.cs file)

Modify the .cs file

When you create a service out of an Oracle E-Business Suite artifact using the Oracle E-Business adapter and want to use it from the Business Data List Web Part in Microsoft Office SharePoint Server, you are expected to provide the full filter clause starting with the WHERE clause. For example, if you want to search for an employee who name is "John", you need to provide the following filter clause in the Business Data List Web Part:

```
where NAME like 'JOHN'
```

However, if you want the user to only provide the name as an input for the filter clause without actually mentioning the whole filter clause, you can add a code in the .cs file that modifies the filter clause coming in from the Business Data List Web Part in Microsoft Office SharePoint Server to pass it to Oracle E-Business in the WHERE clause format.

For example, in the case of this tutorial, if you want the user to enter an employee name in Business Data List Web Part in Microsoft Office SharePoint Server and retrieve record for that employee, you can add the following code in the .cs file:

```
SelectResponse InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE.Select(SelectRequest request)
{
    request.FILTER = "where NAME like '" + request.FILTER + "'"; // The code to avoid the
users from specifying the WHERE clause in the filter from Business Data List Web Part.
    return base.Channel.Select(request);
}
```

Publish the WCF Service

Make sure that SSL is enabled for IIS. For instructions on how to enable SSL for IIS, see <http://go.microsoft.com/fwlink/?LinkId=197170>.

To publish the WCF service:

1. Right-click the project in Solution Explorer, and then click **Publish**.
2. In the **Publish Web** dialog box, specify a URL for the WCF service. For example:

```
https://<COMPUTER_NAME>:<PORT_NUMBER>/MS_SAMPLE_EMPLOYEE/
```

Note

You must publish the WCF service to a SSL-enabled location. In other words, the value in the **Target Location** box must start with "https://". Because the user credentials are passed in the HTTP header, the wizard automatically configured the adapter's binding behavior to use "Transport" as the security mode, which implies SSL encryption. You can of course go back, and edit the web.config file to change the value of the **<security mode>** parameter, but using SSL is always a better option when you have sensitive information transported in clear text in the HTTP header.

3. From the **Copy** box, click **All project files**.
4. Click **Publish**.
5. Verify that the WCF service is published successfully.
 - a. Start the IIS Microsoft Management Console. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services (IIS) Manager**.
 - b. Navigate to the node where you published the service. For the **MS_SAMPLE_EMPLOYEE** service, navigate to **Internet Information Services > <Computer Name> > Web Sites > Default Web Site > MS_SAMPLE_EMPLOYEE**.
 - c. On the right pane, right-click the InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE.svc file, and then click **Browse**.
 - d. The Web page shows up with the URL for retrieving the WSDL. You may want to test metadata retrieval using the **svcutil** command. For example, the command to retrieve metadata for the MS_SAMPLE_EMPLOYEE service is:

```
svcutil.exe https://<COMPUTER_NAME>:<PORT_NUMBER>/MS_SAMPLE_EMPLOYEE/InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE.svc?wsdl
```

Next Step

To create an application definition file for the Oracle E-Business Suite artifact, use the Business Data Catalog Definition Editor. For instructions, see [Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts](#). The application definition file identifies where the LOB data is stored and the format in which it is stored.

See Also

Other Resources

[Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#)

Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts



Time to complete: 15 minutes

Objective: The Business Data Catalog feature in Microsoft SharePoint Server exposes and incorporates data from line-of-business (LOB) applications into portals. To incorporate this data into your portal site, you must build an application definition file that Microsoft Office SharePoint Server can consume.

The Business Data Catalog Definition Editor tool, available with Microsoft Office SharePoint Server 2007 SDK, enables you to create an application definition file for the Business Data Catalog. This tool automatically generates an XML file for the definition file, so you do not need to manually create the file in an XML editor.

The purpose of the Microsoft Office SharePoint Server application that you are creating is to:

- Query for an employee in the MS_SAMPLE_EMPLOYEE interface table using a Business Data List Web Part based on an employee name.
- Perform a full-text search from Microsoft Office SharePoint Server on the MS_SAMPLE_EMPLOYEE interface table.

For each of these requirements, you must complete a set of tasks in the Business Data Catalog Definition Editor tool. This topic provides instructions on how to perform these tasks.

Prerequisites

- Be sure that you have the Business Data Catalog Definition Editor installed as part of the Microsoft Office SharePoint Server 2007 SDK. You can download the SDK from <http://go.microsoft.com/fwlink/?LinkId=104130>.
- Publish the WCF service as described in [Step 1: Use the Oracle E-Business Adapter to Create and Publish a WCF Service](#).

In this topic

We will perform the following tasks in this topic:

1. [Connect to the WCF LOB Service and Create Entity](#)
2. [Specify User Name and Password Headers for the Methods](#)
3. [Scenario 1: Query for Employees using a Business Data List Web Part](#)
4. [Scenario 2: Full-Text Search on MS_SAMPLE_EMPLOYEE Interface Table from Microsoft Office SharePoint Server](#)
5. [Set Default Parameters for the Method Instances](#)
6. [Set up Single Sign-On for Connecting to Oracle E-Business Suite](#)
7. [Export the Application Definition to a File](#)

Connect to the WCF LOB Service and Create Entity

You must connect to the WCF service to extract the Web Services Description Language (WSDL) for the service. From the WSDL, the Business Data Catalog Definition Editor extracts the methods. These methods can be used to create entities. For this tutorial, an entity is created.

To connect to the WCF service and create entities

1. Start the Business Data Catalog Definition Editor. On the **Start** menu, click **Microsoft Business Data Catalog**

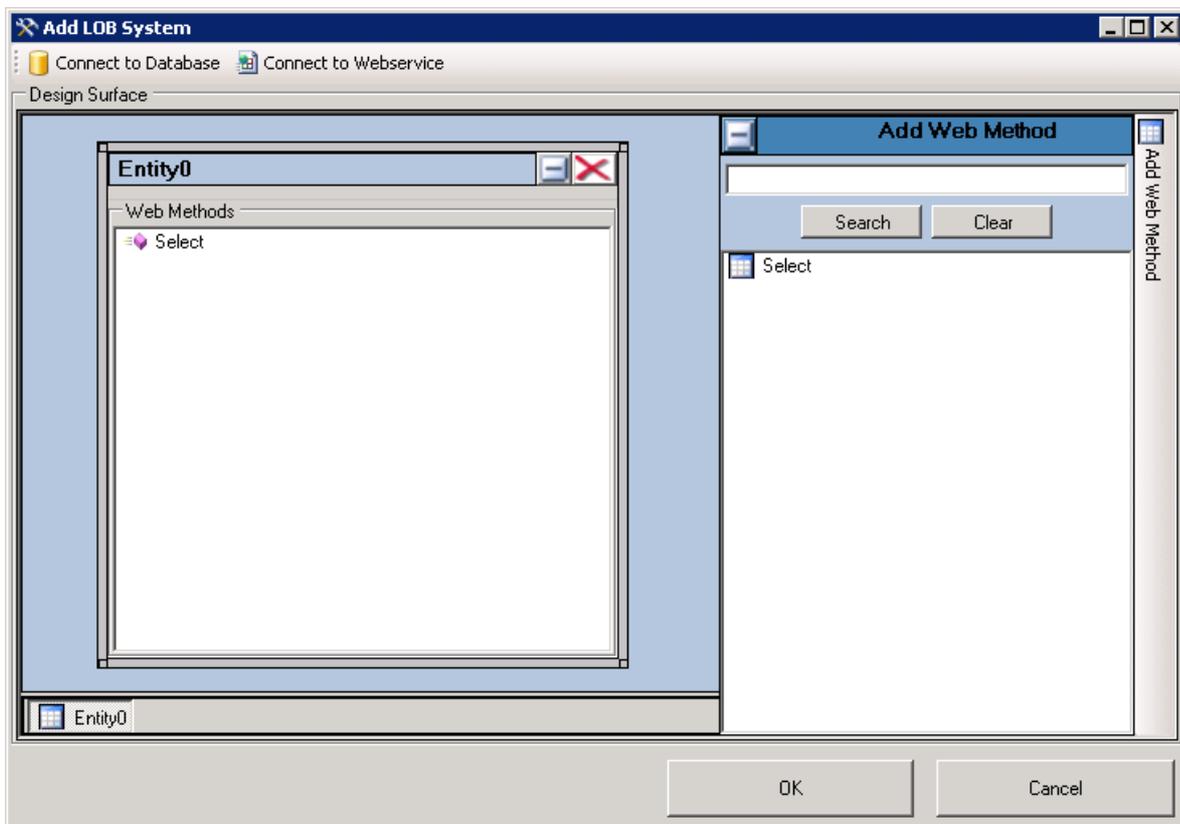
Definition Editor.

2. On the toolbar, click **Add LOB System**.
3. In the Add LOB System window, click **Connect to Webservice**.
4. In the **URL** box, type the URL for the WCF service. For this tutorial, the URL will be:

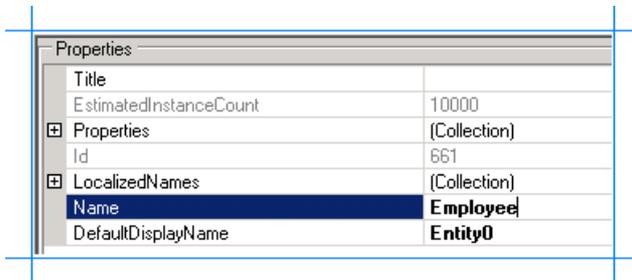
```
https://<COMPUTER_NAME>:<PORT_NUMBER>/MS_SAMPLE_EMPLOYEE/InterfaceTables_FND_APPS_MS_S  
AMPLE_EMPLOYEE.svc
```

The URL is available when you test whether the WCF service is published successfully, as described in [Publish the WCF Service](#).

5. Click **Connect**.
6. To see the operations you selected in the WCF Adapter Service Development Wizard, click the **Add Web Method** tab. You will see the following method: **Select**.
7. Drag the **Select** methods to the Design Surface. As you drag the method to the Design Surface, an entity is created, and the method becomes part of that entity.



8. Click **OK**.
9. In the **Enter the name for the LOB System** dialog box, type a name in the **LOB System Name** box. For this example, call it **MS_SAMPLE_EMPLOYEE**, and then click **OK**.
10. In the Business Data Catalog Definition Editor, the newly created entity is listed as **Entity0**. Rename the entity to **Employee**. Perform the following steps to rename the entity:
 - a. Expand the **MS_SAMPLE_EMPLOYEE** node, and then expand the **Entities** node.
 - b. Select the **Entity0** node.
 - c. In the Properties pane, type **Employee** in the **Name** box.

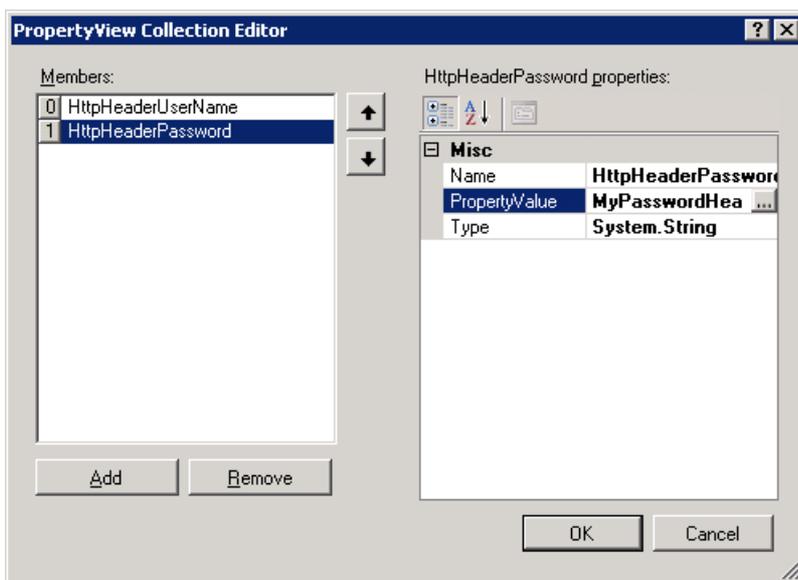


Specify User Name and Password Headers for the Methods

When creating a WCF service for the Select operation on the MS_SAMPLE_EMPLOYEE interface table in Oracle E-Business Suite, you specified user name and password headers as part of the endpoint behavior configuration in [Step 1: Use the Oracle E-Business Adapter to Create and Publish a WCF Service](#). You must specify the same values for the Select method property.

To specify user name and password headers for the Select method

1. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
2. Click the **Select** node, and in the Properties pane click the ellipsis (...) button against the **Properties** box.
3. In the PropertyView Collection Editor window, click **Add**, and in the Property pane, type **HttpHeaderUserName** for the **Name** box. Type **MyUserHeader** for the **PropertyValue** box. Select **System.String** for the **Type** box.
4. In the PropertyView Collection Editor window, click **Add**, and in the Property pane, type **HttpHeaderPassword** for the **Name** box. Similarly, type **MyPasswordHeader** for the **PropertyValue** box. Select **System.String** for the **Type** box.



5. Click **OK**.

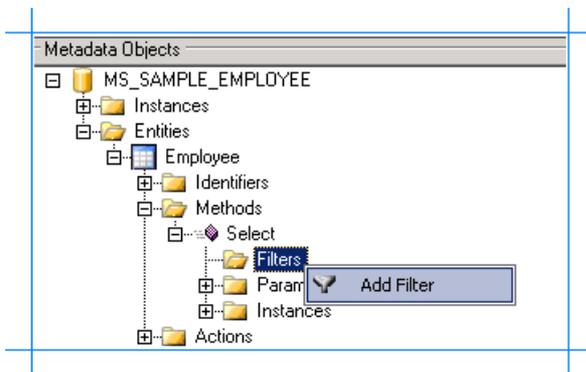
Scenario 1: Query for Employees using a Business Data List Web Part

To create an application definition file that can be used to search for employees from a Business Data List Web Part and based on employee name, you must perform the following set of tasks.

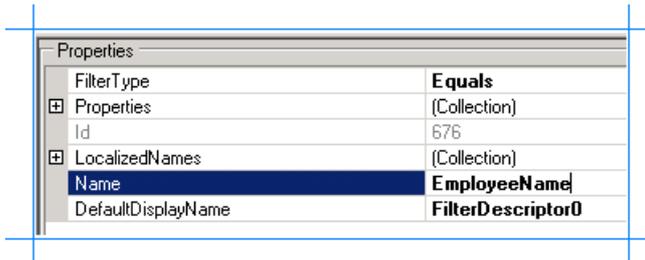
1. In the **Select** method, create a filter and map it to the **FILTER** parameter.
2. Create a **Finder** method instance for the **Select** method. A **Finder** method retrieves a list of records based on a filter.

To create a filter, and map it to the FILTER parameter

1. Create a filter.
 - a. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
 - b. Expand the **Select** method, right-click **Filters**, and then click **Add Filter**.

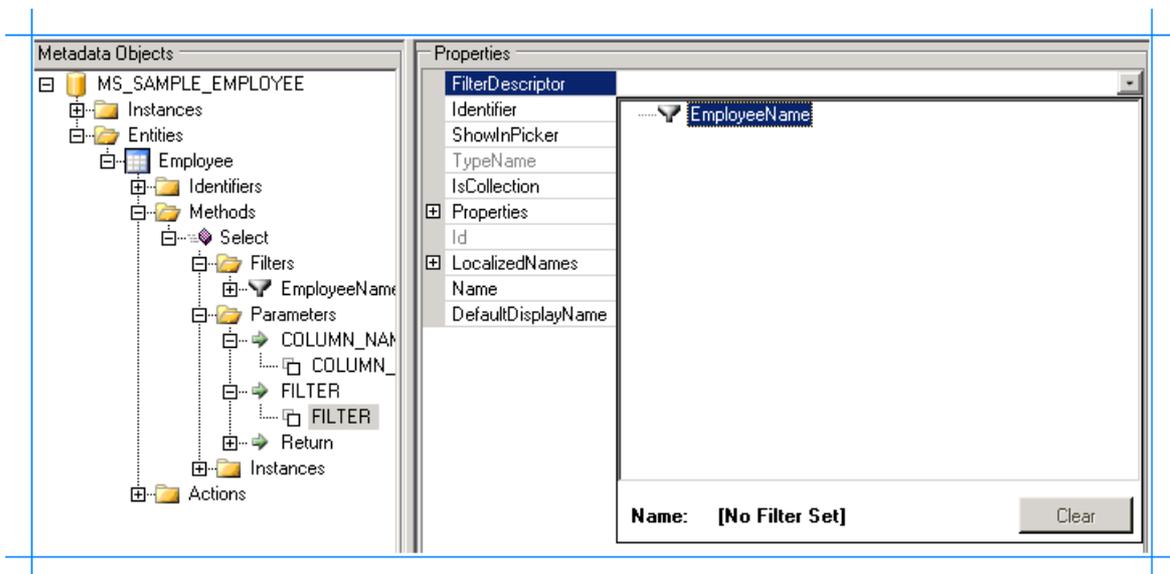


- c. In the Properties pane, for the **FilterType** property, select **Equals**.
- d. In the Properties pane, type **EmployeeName** in the **Name** box.



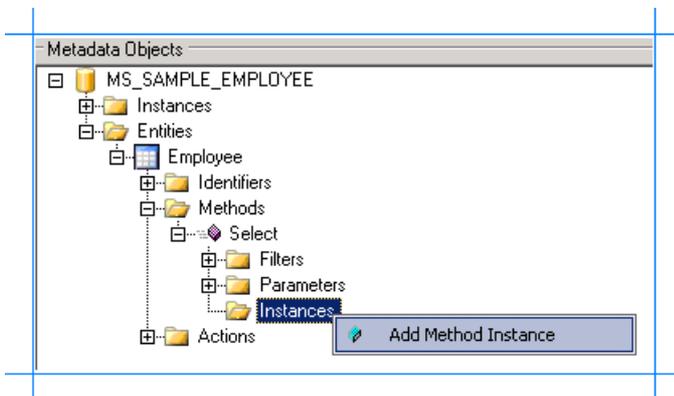
2. Map the filter to the **FILTER** parameter in the **Select** method.

- a. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
- b. Expand the **Select** method, and then expand the **Parameters** node.
- c. Expand the **FILTER** node, and click the second **FILTER** node.
- d. In the Properties pane, select **EmployeeName** from the **FilterDescriptor** list.

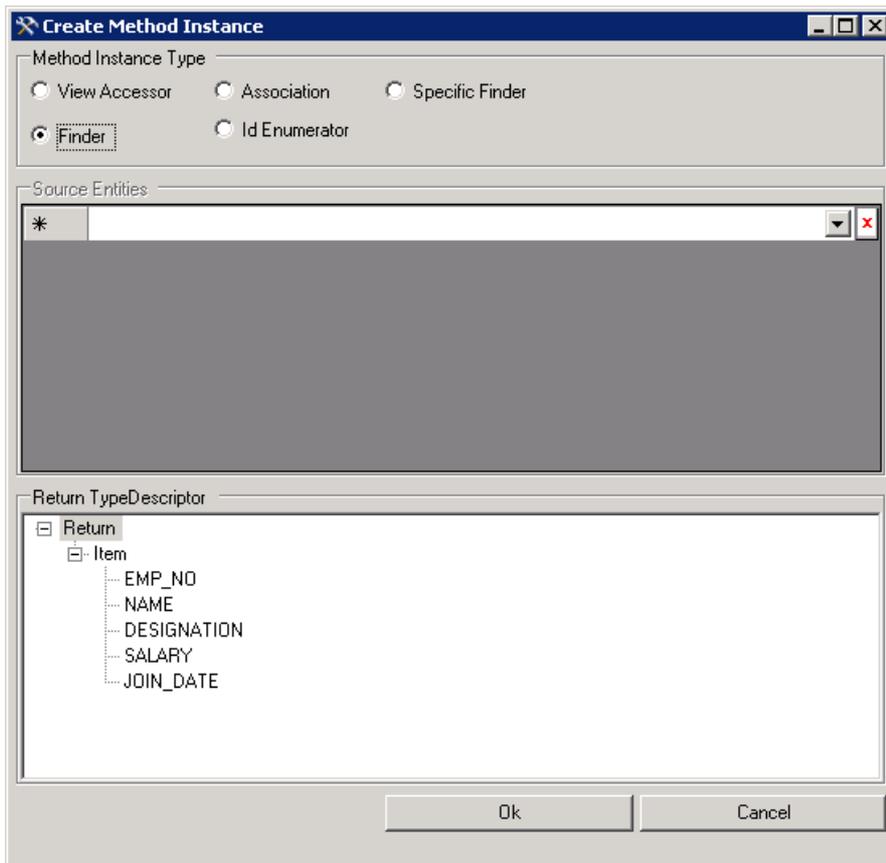


To create a Finder method instance for the Select method

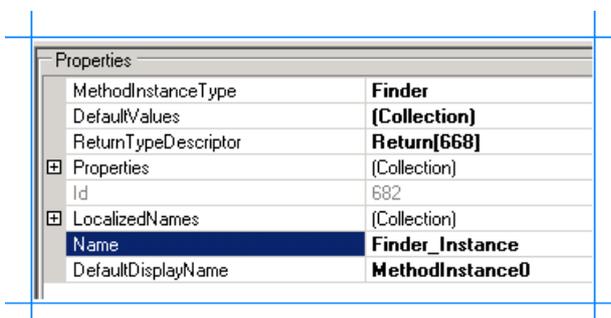
1. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
2. Expand the **Select** node, right-click **Instances**, and then click **Add Method Instance**.



3. In the Create Method Instance window, click **Finder** for **Method Instance Type**. Select **Return** for **Return TypeDescriptor**.



4. Click **OK**.
5. In the Properties pane, type **Finder_Instance** in the **Name** box.



Scenario 2: Full-Text Search on MS_SAMPLE_EMPLOYEE Interface Table from Microsoft Office SharePoint Server

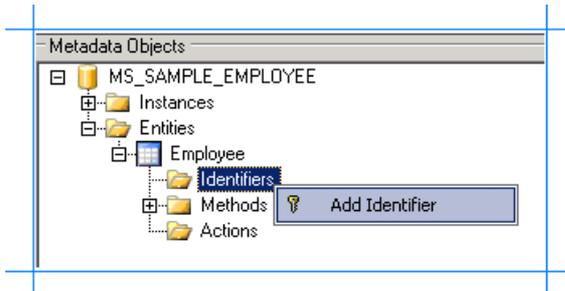
To create an application definition file that can be used to perform a full-text search on MS_SAMPLE_EMPLOYEE interface table from Microsoft Office SharePoint Server, you must perform the following set of tasks.

- In the **Select** method, create an identifier, and map it to the FILTER parameter and the return value that stores the employee name.

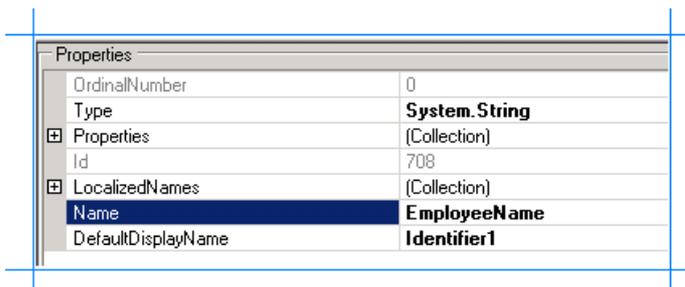
- Create a **Specific Finder** method instance for the **Select**. The **Specific Finder** method will find a specific record based on the identifier, that is, an employee name.
- Create an ID Enumerator method instance.

To create an identifier, and map it to the FILTER parameter and employee name return value

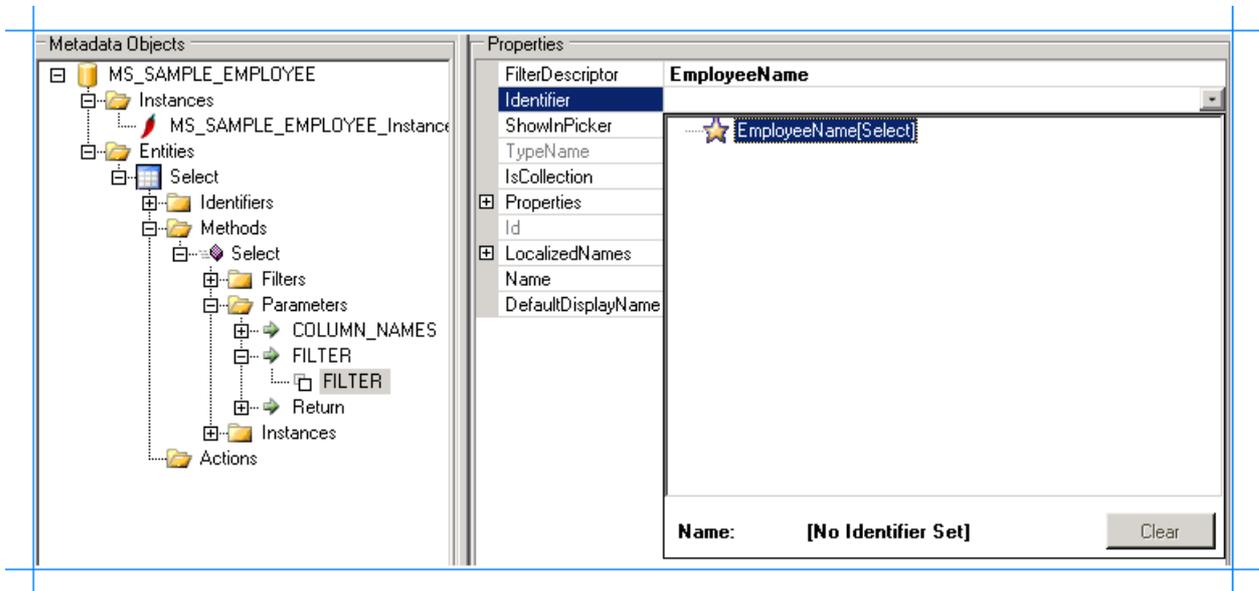
1. Create an identifier for the **Employee** entity.
 - a. In the Metadata Objects pane, expand the **Employee** node.
 - b. Right-click the **Identifiers** node, and then select **Add Identifier**.



- c. In the Properties pane, type **EmployeeName** in the **Name** box.
- d. Select **System.String** for the **Type** box.



2. Map the identifier to the FILTER parameter for the **Select** method.
 - a. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
 - b. Expand the **Select** method, and then expand the **Parameters** node.
 - c. Expand the **FILTER** parameter, and then click the second **FILTER** node.
 - d. In the Properties pane, select **EmployeeName[Employee]** from the **Identifier** list.

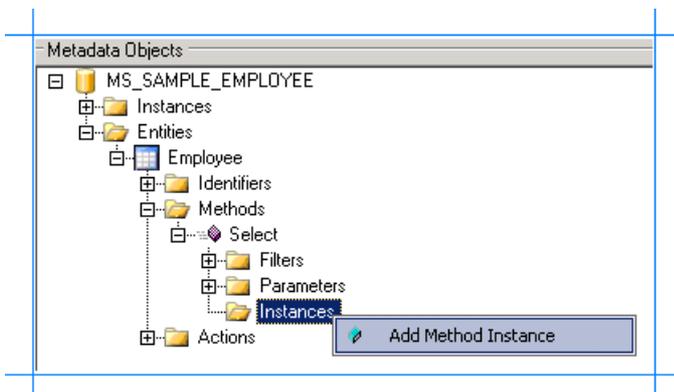


3. Map the identifier to the employee name return value.

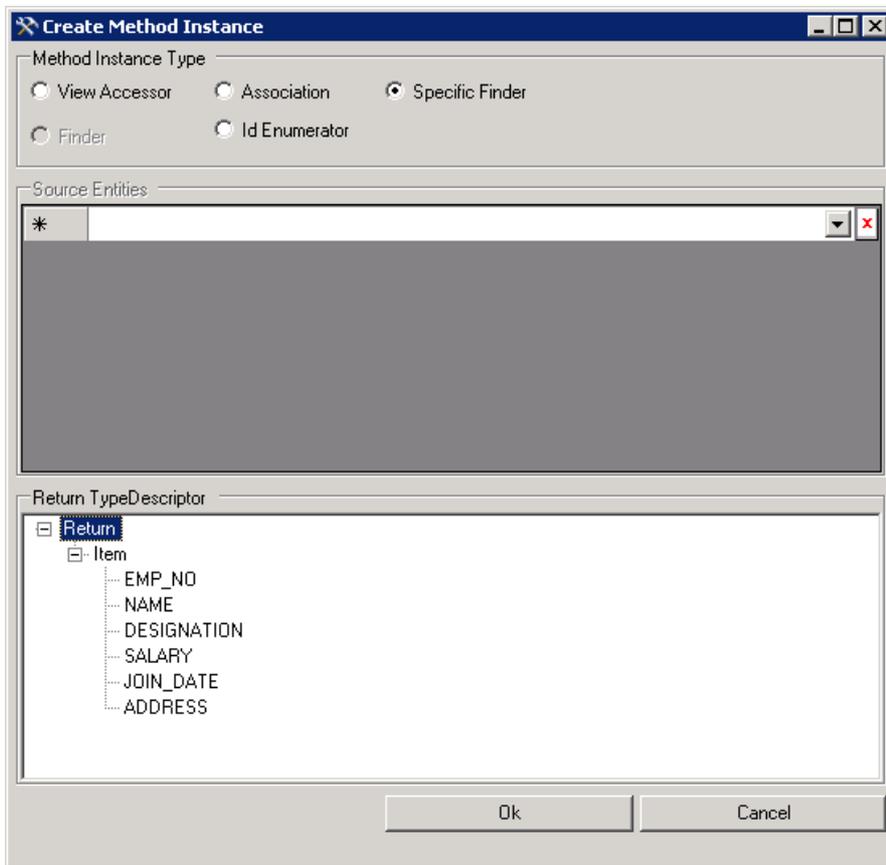
- In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
- Expand the **Select** method, and then expand the **Parameters** node.
- Expand the **Return** node, then the second **Return** node, then the **Item** node, and then click the **Name** node.
- In the Properties pane, select **EmployeeName[Employee]** from the **Identifier** list.

To create a Specific Finder method instance for the Select method

- In the Metadata Objects pane, expand the **Employee** node, and then the **Methods** node.
- Expand the **Select** node, right-click **Instances**, and then select **Add Method Instance** to open the Create Method Instance window.



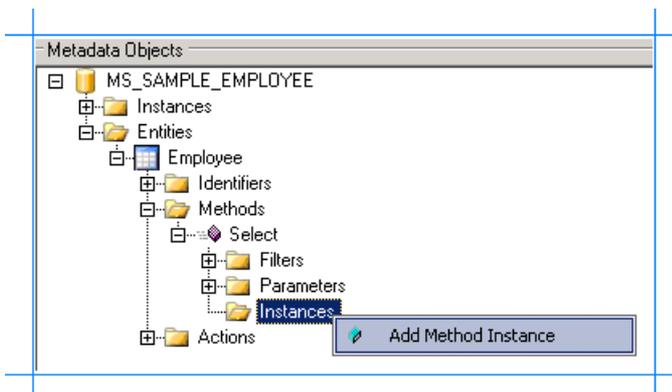
- In the Create Method Instance window, select **Specific Finder** for **Method Instance Type**. Select **Return** for **Return TypeDescriptor**.



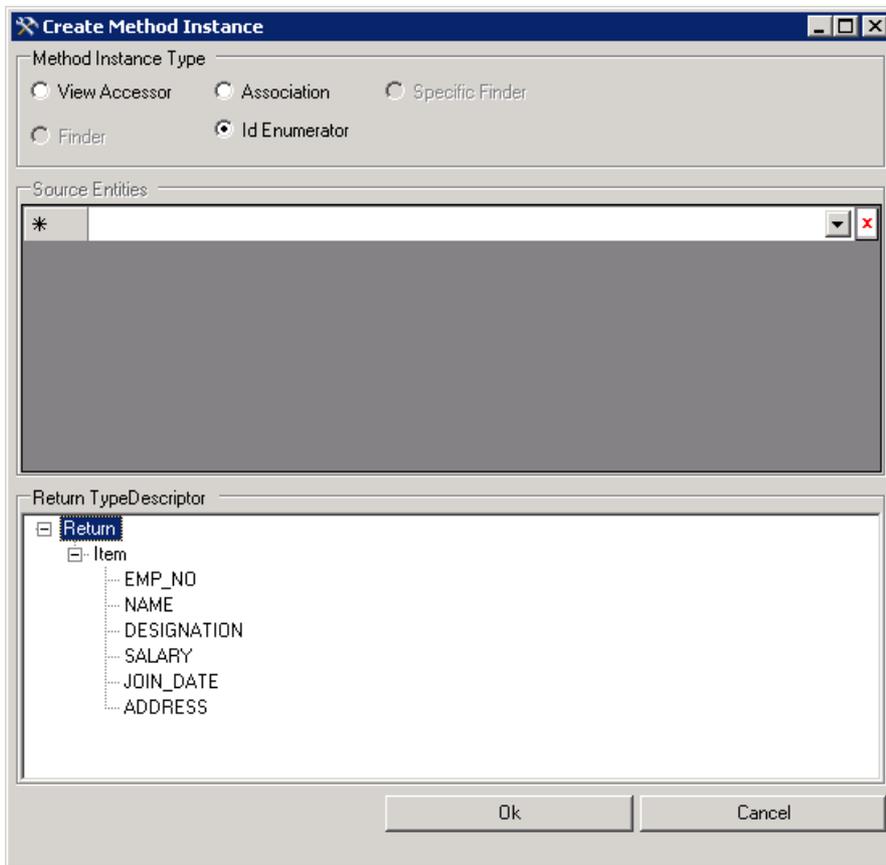
4. Click **OK**.
5. In the Properties pane, type **SpeciFinder_Instance** for the **Name** box.

To create an Id Enumerator method instance for the Select method

1. In the Metadata Objects pane, expand the **Employee** node, and then the **Methods** node.
2. Expand the **Select** node, right-click **Instances**, and then select **Add Method Instance** to open the Create Method Instance window.



3. In the Create Method Instance window, select **Id Enumerator** for **Method Instance Type**. Select **Return** for **Return TypeDescriptor**.



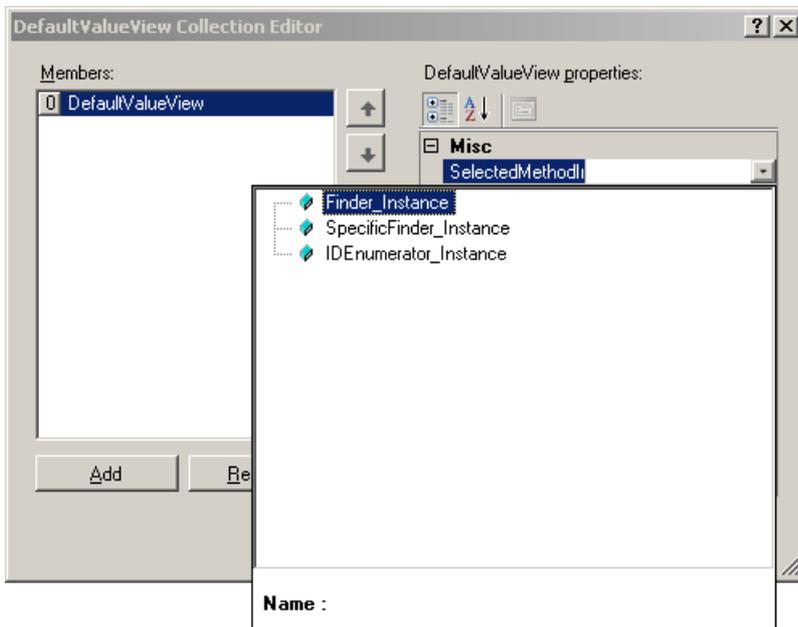
4. Click **OK**.
5. In the Properties pane, type **IDEnumerator_Instance** for the **Name** box.

Set Default Parameters for the Method Instances

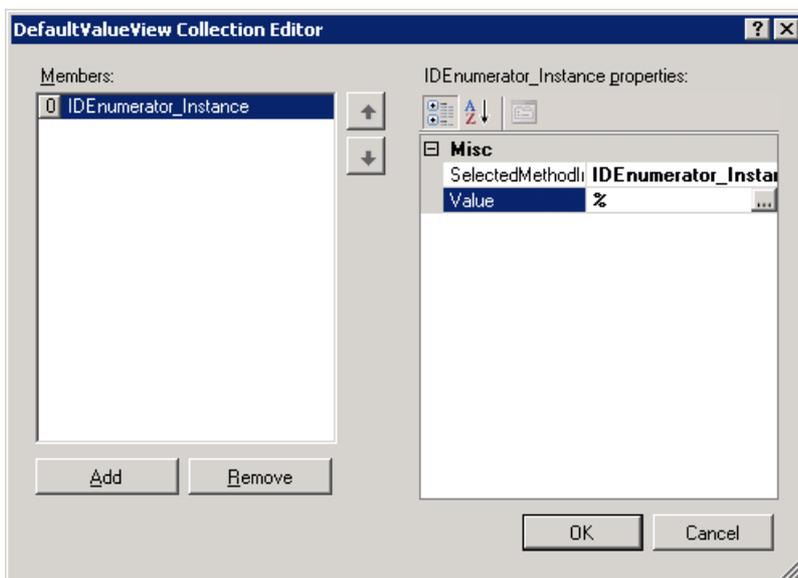
The Select method requires you to specify the column names. Therefore, you need to specify a default value for the **COLUMN_NAMES** parameter for the Finder, Specific Finder, and Id Enumerator method instances created earlier. Additionally, you should also specify a default value for the **FILTER** parameter for the Id Enumerator method instance.

To set the default parameters for the method instances

1. In the Metadata Objects pane, expand the **Employee** node, and then expand the **Methods** node.
2. Expand the **Select** node, and then expand the **Parameters** node.
3. Expand the **COLUMN_NAMES** node, and then select the **COLUMN_NAMES** parameter.
4. In the Properties pane, click the ellipsis button (...) against the **DefaultValues** box.
5. In the **DefaultValueView Collection Editor** dialog box, click **Add**, and in the property pane, click **Finder_Instance** in the **SelectMethodInstance** list.



6. Type * in the **Value** box.
7. Similarly, repeat steps 5 and 6 to add default values for the **SpecificFinder_Instance** and **IDEnumerator_Instance** method instances.
8. In the **DefaultValueView Collection Editor** dialog box, click **OK**.
9. Next, add a default value for the **FILTER** parameter for the **IDEnumerator_Instance** method instance. Expand the **FILTER** node, and then select the **FILTER** parameter.
10. In the Properties pane, click the ellipsis button (...) against the **DefaultValues** box.
11. In the **DefaultValueView Collection Editor** dialog box, click **Add**, and in the property pane, click **IDEnumerator_Instance** in the **SelectMethodInstance** list.
12. Type % in the **Value** box.



13. In the **DefaultValueView Collection Editor** dialog box, click **OK**.

Set up Single Sign-On for Connecting to Oracle E-Business Suite

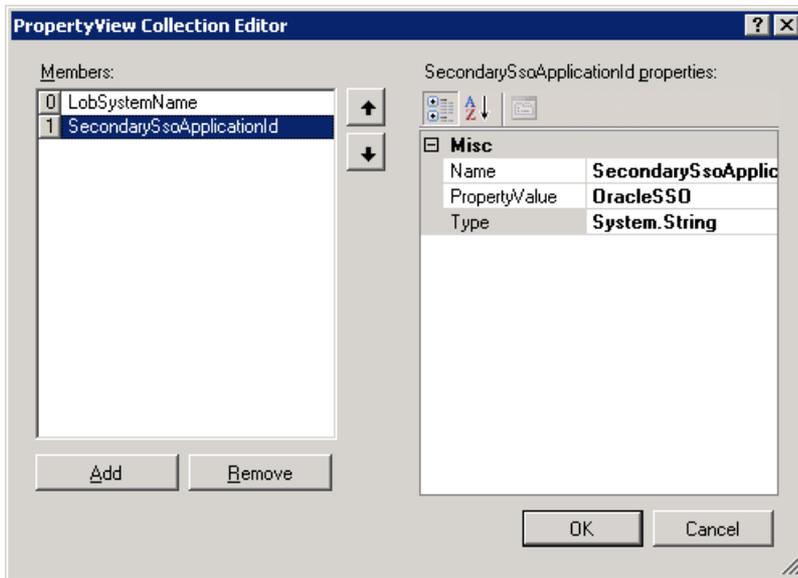
After you have finished performing all the procedures in this topic, you will have created an application definition file that can be imported into a SharePoint application. From the application, you invoke the methods to retrieve relevant data from Oracle E-Business Suite. To enable this, you must create a mapping between a user in the Oracle E-Business Suite and the user in the SharePoint application. You create this mapping in SharePoint Central Administration console after you have imported the application definition file.

However, to create the mapping you must set a property **SecondarySsoApplicationId** in the Business Data Catalog Definition

Editor.

To set the SecondarySsoApplicationId property

1. In the Metadata Objects pane, expand the **MS_SAMPLE_EMPLOYEE** node, and then expand the **Instances** node.
2. Click **MS_SAMPLE_EMPLOYEE_Instance**, and in the Properties pane, click the ellipsis (...) button against the **Properties** box.
3. In the **PropertyView Collection Editor** dialog box, click **Add**, and in the Property pane, type **SecondarySsoApplicationId** for the **Name** box. Similarly, type **OracleSSO** for the **PropertyValue** box. Select **System.String** for the **Type** box.



4. Click **OK**.

Export the Application Definition to a File

You have now created an application definition that contains Oracle E-Business Suite instance metadata. You must export this definition to an XML file, which can be imported into Microsoft Office SharePoint Server.

To export the application definition to a file

1. In the Metadata Objects pane, right-click the **MS_SAMPLE_EMPLOYEE** node, and then click **Export**.
2. Save the file as Employee.xml.

Next Steps

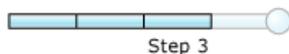
You must now create a SharePoint application to retrieve data from Oracle E-Business Suite. For instructions, see [Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite](#).

See Also

Other Resources

[Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#)

Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite



Time to complete: 15 minutes

Objective: You must now import the application definition file in Microsoft Office SharePoint Server, and set up an application to retrieve data from Oracle E-Business Suite.

Prerequisites

- You should have created an application definition file as described in [Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts](#).
- The Microsoft Single Sign-on service must be running.

In this topic

We will perform the following tasks in this topic:

- [Creating an SSO Application in SharePoint](#)
- [Creating a Shared Services Provider](#)
- [Importing the Application Definition File](#)

Creating an SSO Application in SharePoint

To access the data in Oracle E-Business Suite from a SharePoint application, you must set up an SSO application that maps a SharePoint user to an Oracle E-Business Suite user. Creating an SSO application in SharePoint involves the following steps:

1. **Manage server settings for single sign-on.** In this step, you specify a user account that can manage and set up the single sign-on service. You can do so on the Manage Server Settings page. This option is available from the SharePoint Central Administration console. For more information about this step, refer to the "Configure Single Sign-On for Office SharePoint Server 2007" section at <http://go.microsoft.com/fwlink/?LinkId=105291>.
2. **Manage settings for enterprise application definitions.** In this step, you configure the settings for the enterprise application definition. You can do so from the Manage Settings for Enterprise Application Definitions page. This option is available from the SharePoint Central Administration console.
 - a. On Central Administration, on the top navigation bar, click **Operations**.
 - b. On the Operations page, in the **Security Configuration** section, click **Manage settings for single sign-on**.
 - c. On the Manage Settings for Single Sign-On page, in the **Enterprise Application Definition Settings** section, click **Manage settings for enterprise application definitions**.
 - d. On the Manage Enterprise Application Definitions page, provide values for the **Display name**, **Application name**, and the **Contact e-mail address** fields.

◆ Important

For the **Application name** field, make sure you specify the same SSO application name that you specified for the **SecondarySsoApplicationId** variable while creating the application definition file, as described in [Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts](#).

e. Leave the other fields as default, and click **OK**.

3. **Manage account information for enterprise application definitions.** In this step, you enable individual users or groups to connect to an enterprise application from SharePoint. Essentially, in this step you map an individual user or group to a user in the LOB system. You also specify the credentials to connect to the LOB system. You can do so from the Manage Account Information for Enterprise Application Definitions page. This option is available from the SharePoint Central Administration console. For more information about this step, refer to the "Manage account information for an enterprise application definition" section at <http://go.microsoft.com/fwlink/?LinkId=105291>.

Creating a Shared Services Provider

A Shared Service Provider is a logical grouping of shared services and their supporting resources. You can create an SSP by using the SharePoint Central Administration console.

You must define a Web site when creating an SSP. Remember the port number and the site address that you create. You will import the Business Data Catalog application definition to this site.

For more information about creating an SSP, see "Chapter overview: Create and configure Shared Services Providers" at <http://go.microsoft.com/fwlink/?LinkId=105119>.

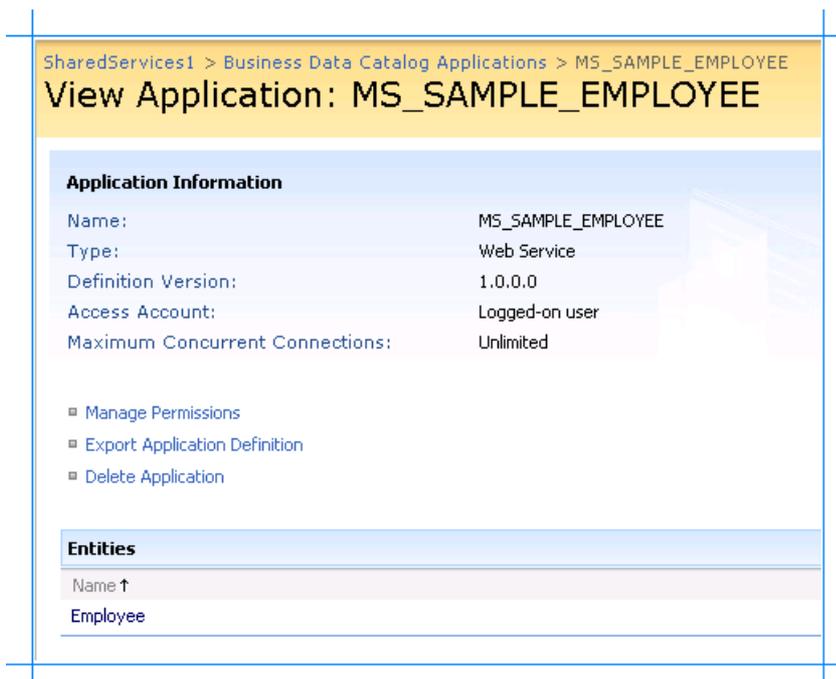
Importing the Application Definition File

You must now import the application definition file into the SSP.

To import the application definition file

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP to which you want to import the application definition.
3. In the **Business Data Catalog** section, click **Import application definition**.
4. On the Import Application Definition page that opens, browse to Employee.xml, select the file, and then click **Open**.
5. Click **Import**.
6. After the application definition file is imported successfully, click **OK**.

The application created as a result of importing the application definition file, MS_SAMPLE_EMPLOYEE, appears.



Next Steps

Now, you are ready to create Web Parts to create a SharePoint site to view and search the business data that will be extracted from Oracle E-Business Suite. We will create a:

- Business Data List Web Part to display employee records from the MS_SAMPLE_EMPLOYEE interface table. See [Scenario 1: Display Data Using Business Data List Web Part](#).
- Search Box Web Part to perform a full-text search on the MS_SAMPLE_EMPLOYEE interface table. See [Scenario 2: Perform a Search Using the Search Box Web Part](#).

See Also

Other Resources

[Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#)

Scenario 1: Display Data Using Business Data List Web Part

We will use the **Business Data List** Web Part for the **Finder** method instance. This Web Part enables you to specify a search expression to retrieve a list of employees from Oracle E-Business Suite. For this tutorial, this is called the Display Employees Web Part. This section provides instructions to create this Web Part. For more information about creating Web Parts, see "Customize business data lists, Web Parts, and sites" at <http://go.microsoft.com/fwlink/?LinkId=104131>.

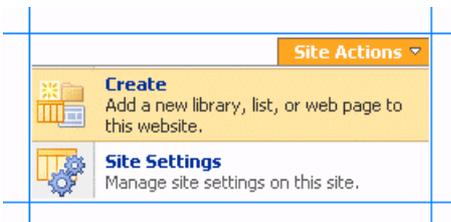
You must create a Web Part page before adding the Web Parts.

Creating a Web Part Page

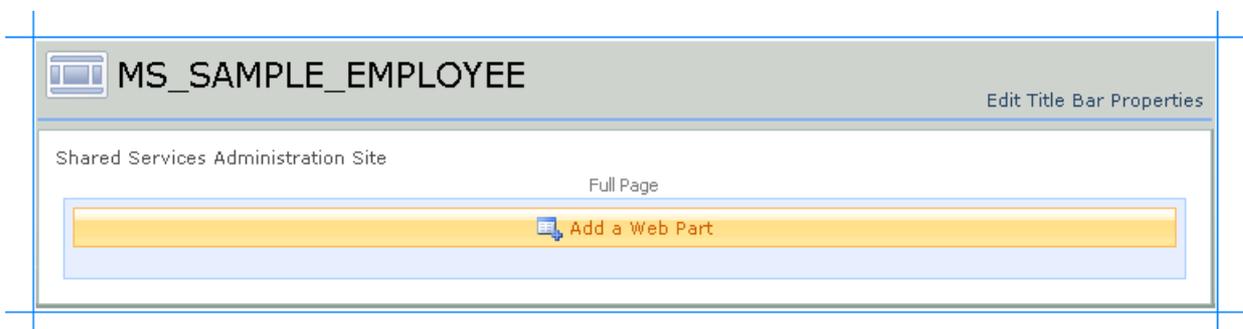
This section provides instructions to create a Web Part page.

To create a Web Part page

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP to which you want to import the application definition.
3. On the Shared Services Administration page, in the upper-right corner, click **Site Actions**, and then click **Create**.



4. On the Create page, in the **Web Pages** section, click **Web Part Page**.
5. On the New Web Part page, do the following:
 - a. In the **Name** field, type a name for the page. For this tutorial, type the name as **MS_SAMPLE_EMPLOYEE**.
 - b. Select the **Overwrite if file already exists** check box, if you want to overwrite old pages with the same name as the new page you create.
 - c. In the **Layout** section, from the **Choose a Layout Template** box, select a layout for the Web Part page. For this tutorial, select **Full Page, Vertical**.
 - d. In the **Save Location** section, in the **Document Library** list, click **Form Templates**.
 - e. Click **Create**. The following figure shows the Web Part page after it is created.



- f. You must now add the Web Parts to this page.

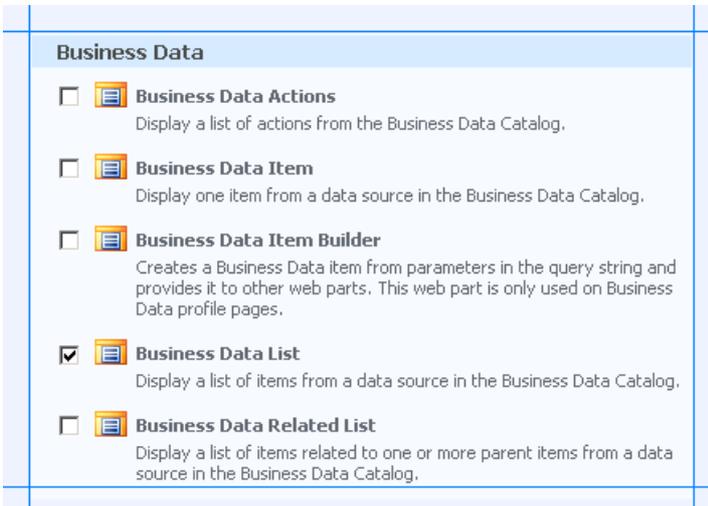
Adding a Business Data List Web Part

You must now add a Business Data List Web Part to the Web Part page. Using this Web Part you will retrieve a list of employee records from the MS_SAMPLE_EMPLOYEE interface table in Oracle E-Business Suite that matches a search expression. This Web Part corresponds to the **Finder** method instance (*Finder_Instance*) that you created in the Business Data Catalog

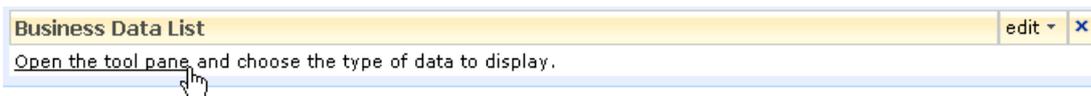
Definition Editor.

To add a Business Data List Web Part

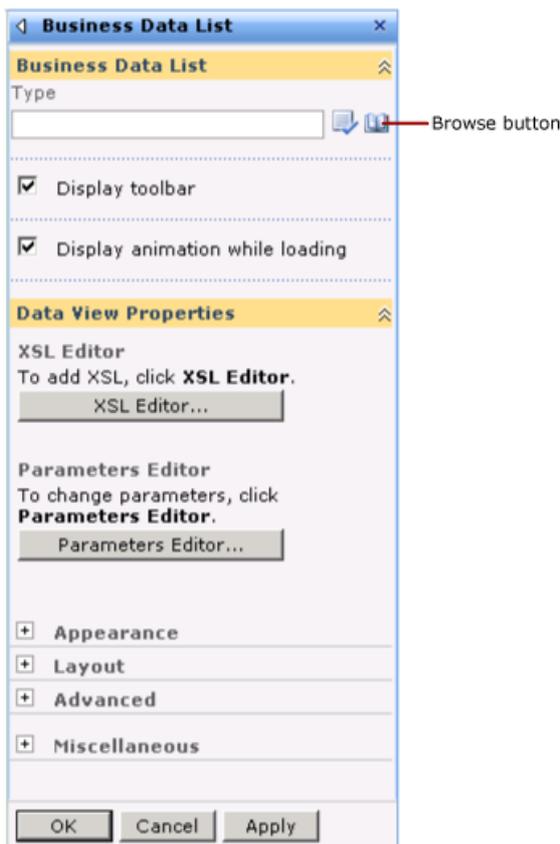
1. On the MS_SAMPLE_EMPLOYEE page, click **Add a Web Part**.
2. In the **Add Web Parts** dialog box, in the **Business Data** section, select the **Business Data List** check box, and then click **Add**.



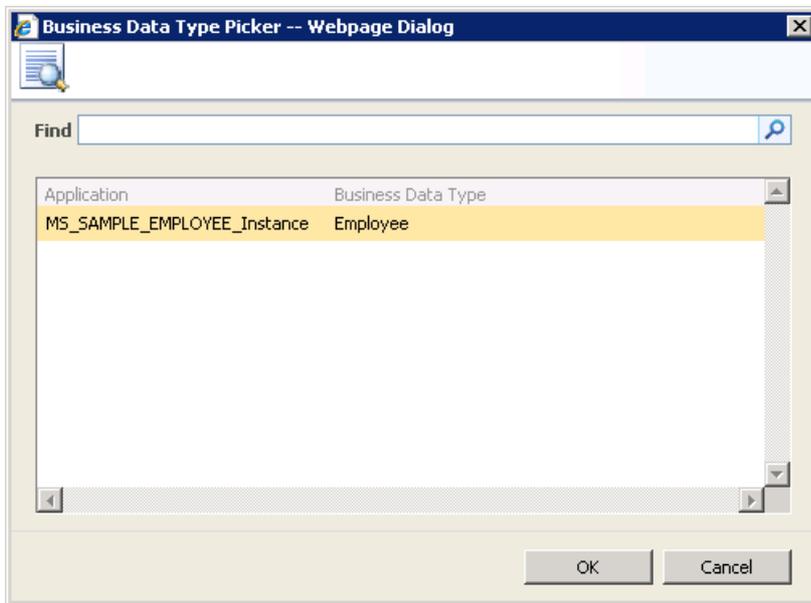
3. In the newly added Business Data List Web Part, click the **Open the tool pane** link.



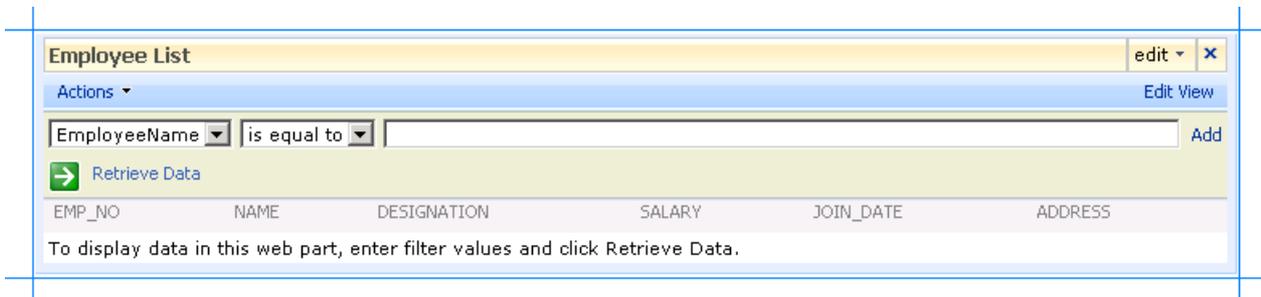
4. The Business Data List tool pane opens in the right pane. In the **Business Data List** section, for the **Type** field, click the **Browse** button.



5. In the **Business Data Type Picker** dialog box, select the **MS_SAMPLE_EMPLOYEE_Instance** application, and then click **OK**.



6. Expand the **Appearance** node, and in the **Title** box, type a title for the Web Part. For this Web Part, type **Employee List**.
7. In the Business Data List tool pane, click **Apply**, and then click **OK**. The Business Data List Web Part now looks like the following:



8. The Web Part lists the fields that are returned by executing the Select operation on the MS_SAMPLE_EMPLOYEE interface table.

See Also

Tasks

[Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite](#)

Concepts

[Scenario 2: Perform a Search Using the Search Box Web Part](#)

Scenario 2: Perform a Search Using the Search Box Web Part

We will configure the search settings in Microsoft Office SharePoint Server to configure a search application using which you can perform a full text search on the MS_SAMPLE_EMPLOYEE interface table in Oracle E-Business Suite. Later, we will add a Search Box Web Part to from where you can perform the search.

Configuring search in Microsoft Office SharePoint Server involves the following:

1. [Define the Content Source](#)
2. [Define a Scope for the Crawled Content](#)
3. [Add the Scope to the Search Dropdown](#)
4. [Add the Search Box Web Part](#)

Define the Content Source

This section talks about defining a content source from where Microsoft Office SharePoint Server can crawl the data. This involves mapping the content to the Id Enumerator method instance created in

[Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts.](#)

To define a content source

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the Shared Service Provider (SSP) where you want to configure the search application.
3. On the Home page, in the **Search** section, click **Search settings**.
4. On the Configure Search Settings page, in the left pane under **Crawling**, click **Default content access account** to specify an account to use as the default account when crawling content.
5. On the Default Content Access Account page, specify the user name and password credentials, and click **OK**. You will return to the Search Administration page.
6. In the left pane under **Crawling**, click **Content Sources**.
7. On the Manage Content Sources page, click **New Content Source**.
8. On the Manage Content Sources page, click **New Content Source**.
9. On the Add Content Source page:
 - a. Type **MS_SAMPLE_EMPLOYEE** in the **Name** box.
 - b. In the **Content Source Type** area, click **Business Data**.
 - c. In the **Applications** area, click **Crawl selected applications**, and then select the **MS_SAMPLE_EMPLOYEE_Instance** check box.
 - d. In the **Start Full Crawl** area, select the **Start full crawl of this content source** check box, and then click **OK**.

Add Content Source

Use this page to add a content source.

* Indicates a required field

Name

Type a name to describe this content source.

Name: *

MS_SAMPLE_EMPLOYEE

Content Source Type

Select what type of content will be crawled.

Note: This cannot be changed after this content source is created since other settings depend on it.

Select the type of content to be crawled:

- SharePoint Sites
- Web Sites
- File Shares
- Exchange Public Folders
- Business Data

Applications

Select whether to crawl all applications in the Business Data Catalog, or include only selected applications in this content source.

An application may be included in only one Business Data content source.

Select the Business Data Catalog applications to be crawled:

- Crawl entire Business Data Catalog
- Crawl selected applications
- MS_SAMPLE_EMPLOYEE_Instance

Crawl Schedules

Select the crawl schedules for this content source.

Select the schedule that this should be a part of:

Full Crawl

None

[Create schedule](#)

Incremental Crawl

None

[Create schedule](#)

Start Full Crawl

Select "Start full crawl of this content source" and click "OK" to start a full crawl of this content source.

Start full crawl of this content source

OK

Cancel

10. You will return to the Manage Content Sources page with the new content source added. The content source will crawl through the data in the MS_SAMPLE_EMPLOYEE interface table in the Oracle E-Business Suite. Wait until the crawling is completed.

11. In the left pane under **Crawling**, click **Crawl Log**, and then verify the log file to ensure that the crawling is successful.

Define a Scope for the Crawled Content

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the Shared Service Provider (SSP) where you want to configure the search application.
3. On the Home page, in the **Search** section, click **Search settings**.
4. On the Configure Search Settings page, in the left pane under **Queries and Results**, click **Scopes** to define a scope for the crawling of data.
5. On the View Scopes page, click **New Scope**.
6. On the Create Scope page, type **MS_SAMPLE_EMPLOYEE_Search** in the **Title** box, and then click **OK**.

Create Scope

Title and Description

Type a unique title and description for your scope. The title is displayed in the search dropdown, search results pages, advanced search, and elsewhere.

Title: *

MS_SAMPLE_EMPLOYEE_Search

Description:

Last modified by:

EXAMPLE\administrator

Target Results Page

Specify a specific search results page to send users to for results when they search in this scope, or choose to use the default.

Use the default Search Results Page

Specify a different page for searching this scope

Target results page: *

OK

Cancel

7. You will return to the View Scopes page with the new scope added. In the **Update Status** column for the newly added scope, click the **Add rules** link.
8. On the Add Scope Rule page:
 - a. In the **Scope Rule Type** area, click **Content Source**.
 - b. In the **Content Source** list, click **MS_SAMPLE_EMPLOYEE**, and then click **OK**.

Add Scope Rule

Scope Rule Type

Scope rules define what is in or not in a scope. Use different types of rules to match items in various ways

Web Address (http://server/site)

Property Query (Author = John Doe)

Content Source

All Content

Content Source

MS_SAMPLE_EMPLOYEE

Behavior

Decide how this rule should be applied to the overall scope. The scope-wide filter is used when combining the items matching all rules to determine what is in the scopes overall.

Include - Any item that matches this rule will be included, unless the item is excluded by another rule

Require - Every item in the scope must match this rule

Exclude - Items matching this rule will be excluded from the scope

OK

Cancel

9. You will return to the View Scopes page with the rule added for the scope. In the left pane, click **Search Administration**.
10. On the Search Administration page, locate the **Scopes needing update** row, and click the **Start update now** link.

The **Scope update status** row will display the status of the scope update. Wait until the update is complete. After the update is completed, the scope is ready to be used.

Add the Scope to the Search Dropdown

After you have created the search scope, you must add the scope to the search dropdown in Microsoft Office SharePoint Server so that it can be used.

To add the scope to the search dropdown

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the Shared Service Provider (SSP) where you want to configure the search application.
3. On the Shared Services Administration page, in the upper-right corner, click **Site Actions**, and then click **Site Settings**.
4. On the Site Settings page, in the **Site Collection Administration** section, click **Search scopes**.
5. On the View Scopes page, click the **Search Dropdown** link.

View Scopes

 This feature is not available for Shared Service Provider admin sites.

Use this page to view and manage search scopes. The order in which the search scopes appear in this list is the order in which they will appear in the search scope list next to the Search box.

Title	Update Status	Shared	Items
Display Group: Search Dropdown (2)			
All Sites	Ready	<input checked="" type="checkbox"/>	26
People	Ready	<input checked="" type="checkbox"/>	0
Display Group: Advanced Search (1)			
All Sites	Ready	<input checked="" type="checkbox"/>	26
Unused Scopes (1)			
MS_SAMPLE_EMPLOYEE_Search	Ready	<input checked="" type="checkbox"/>	10

6. On the Edit Scope Display Group page:

- In the **Scopes** area, select the **MS_SAMPLE_EMPLOYEE_Search** check box.
- In the **Default Scope** area, click **MS_SAMPLE_EMPLOYEE_Search** in the **Default Scope** list, and then click **OK**.

Edit Scope Display Group

Title and Description Type a title and description for your new display group.	Title: * Search Dropdown Description: Scopes in this group will appear with the search box in the scopes dropdown.												
Scopes Select or clear the check box next to each scope you want to include in this display group. To specify the order of the scopes, select a number in the Position from Top box.	<table border="1"><thead><tr><th>Display</th><th>Scope Name</th><th>Position from Top</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/></td><td>All Sites</td><td>1</td></tr><tr><td><input checked="" type="checkbox"/></td><td>People</td><td>2</td></tr><tr><td><input checked="" type="checkbox"/></td><td>MS_SAMPLE_EMPLOYEE_Search</td><td>3</td></tr></tbody></table>	Display	Scope Name	Position from Top	<input checked="" type="checkbox"/>	All Sites	1	<input checked="" type="checkbox"/>	People	2	<input checked="" type="checkbox"/>	MS_SAMPLE_EMPLOYEE_Search	3
Display	Scope Name	Position from Top											
<input checked="" type="checkbox"/>	All Sites	1											
<input checked="" type="checkbox"/>	People	2											
<input checked="" type="checkbox"/>	MS_SAMPLE_EMPLOYEE_Search	3											
Default Scope	Default Scope MS_SAMPLE_EMPLOYEE_Search												

7. You will return to the View Scopes page with the MS_SAMPLE_EMPLOYEE_Search scope added in the Search Dropdown display group.

Add the Search Box Web Part

To enable the users to perform a full-text search on the MS_SAMPLE_EMPLOYEE interface table in Oracle E-Business Suite, you must now create a Web part page, and add a Search Box Web Part to it.

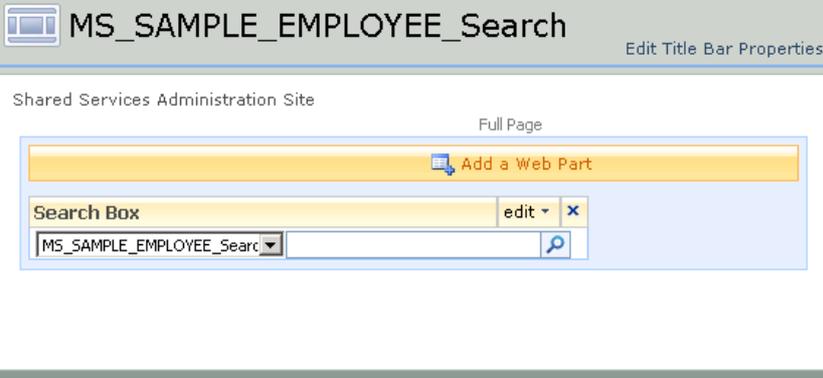
To add the Search Box Web Part

- Create a Web Part page called **MS_SAMPLE_EMPLOYEE_Search**. To know the steps for creating a Web Part page, see [To create a Web Part page in Scenario 1: Display Data Using Business Data List Web Part](#).
- On the MS_SAMPLE_EMPLOYEE_Search page, click **Add a Web Part**.
- In the **Add Web Parts** dialog box, in the **Search** section, select the **Search Box** check box, and then click **Add**.

Search

-  **Advanced Search Box**
Used for parameterized searches based on properties and combinations of words.
-  **Federated Results**
This webpart displays the Federated results from the configured location.
-  **People Search Box**
Used to search people.
-  **People Search Core Results**
This web part displays the people search results and the properties associated with them.
-  **Search Action Links**
Web part to display the search action links.
-  **Search Best Bets**
Web part to display the special term and high confidence results.
-  **Search Box**
Used to search document and items.
-  **Search Core Results**
This web part displays the search results and the properties associated with them.

4. The Search Box Web part is added to the MS_SAMPLE_EMPLOYEE_Search page.



See Also

Tasks

[Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite](#)

Concepts

[Scenario 1: Display Data Using Business Data List Web Part](#)

Step 4: Test Your SharePoint Application



Time to complete: 10 minutes

Objective: After you have added Web Parts in the SharePoint site and created an application, you must test the application by retrieving some data from the Oracle E-Business Suite. This topic provides instructions on how to use the application to retrieve the data from the Oracle E-Business Suite.

Prerequisites

You should have created the Web Part page that contains the appropriate Web Parts to retrieve business data. See [Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite](#).

Scenario 1: To test the SharePoint application created using Business Data List Web Part

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP under which you created the application.
3. In the left pane, click **View All Site Content**. In the right pane, click **Form Templates**.
4. In the **Form Category** list, click **MS_SAMPLE_EMPLOYEE**. You specified this name when you created the Web Part page in [Scenario 1: Display Data Using Business Data List Web Part](#).
5. Search for employees based on a search string. For example, to search for all the employees, type **%** in the text box, and click **Retrieve Data**. The following figure shows the records retrieved from Oracle E-Business Suite:

MS_SAMPLE_EMPLOYEE

Shared Services Administration Site

Employee List

Actions

EmployeeName is equal to % Add

Retrieve Data

EMP_NO	NAME	DESIGNATION	SALARY	JOIN_DATE	ADDRESS
10002	JEFF PRICE	MANAGER	25000	2007-12-14T18:30:00Z	8888 Right St, Redmond 96314
10003	DON HALL	ACCOUNTANT	12000	2005-10-28T18:30:00Z	4567 Main St, New York 98052
10004	KEITH HARRIS	SUPERVISOR	15000	2004-01-10T18:30:00Z	2345 Second St, Los Angeles 90000
10005	JIM HANCE	ACCOUNTANT	13500	2008-02-12T18:30:00Z	2222 Central Bay, New Jersey 92213
10006	ANDY JACOBS	DEVELOPER	14000	2005-04-19T18:30:00Z	9876 Central Place, Redmond 96315
10007	JAY HAMLIN	ACCOUNTANT	25000	2007-11-13T18:30:00Z	2244 Main St, New Jersey 95313
10008	DAVID PELTON	DEVELOPER	12000	2006-05-18T18:30:00Z	7645 Down St, New York 98153
10009	BRIAN PERRY	MANAGER	15000	2004-06-10T18:30:00Z	1111 Central St, New York 98153

6. You can also search for a specific employee by entering their first name or last name:
 - To search using the first name, type the initial letters of an employee name followed by the **%** symbol to return records of all the employees matching the search criteria.
 - To search using the last name, type **%** followed by employee's last name.

Note

The search string is case sensitive.

MS_SAMPLE_EMPLOYEE

Shared Services Administration Site

Employee List

Actions ▾

EmployeeName ▾ is equal to ▾ KEITH%

EMP_NO	NAME	DESIGNATION	SALARY	JOIN_DATE	ADDRESS
10004	KEITH HARRIS	SUPERVISOR	15000	2004-01-10T18:30:00Z	2345 Second St, Los Angeles 90000

Scenario 2: To test the SharePoint application created to perform a full-text search

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP under which you created the application.
3. In the left pane, click **View All Site Content**. In the right pane, click **Form Templates**.
4. In the **Form Category** list, click **MS_SAMPLE_EMPLOYEE_Search**. You specified this name when you created the Web Part page in [Add the search-related Web Parts in Scenario 2: Perform a Search Using the Search Box Web Part](#).
5. The MS_SAMPLE_EMPLOYEE_Search page displays the search box where you can perform a full-text search on the MS_SAMPLE_EMPLOYEE table. For example, if you want to search for all the employees who live in New York, type **New York** in the search box, and press ENTER.

MS_SAMPLE_EMPLOYEE_Search

Shared Services Administration Site

MS_SAMPLE_EMPLOYEE_Search ▾ New York

6. A page appears with the search results. Each matching records is displayed as a link in the search results page.

Shared Services Administration: SharedServices2

Home

Site Actions ▾

This Site: Shared Services Adm ▾ New York

Results by Relevance | [View by Modified Date](#) | [Alert Me!](#) [RSS](#)

Results 1-3 of 3. Your search took 0.06 seconds.

[Employee.aspx](#)

7645 Down St, **New York** 98153

<http://csdedg-2plh7/ssp/admin/Content/Employee.aspx?NAME=DAVID+PELTON> - 1/29/2009

[Employee.aspx](#)

4567 Main St, **New York** 98052

<http://csdedg-2plh7/ssp/admin/Content/Employee.aspx?NAME=DON+HALL> - 1/29/2009

[Employee.aspx](#)

1111 Central St, **New York** 98153

<http://csdedg-2plh7/ssp/admin/Content/Employee.aspx?NAME=BRIAN+PERRY> - 1/29/2009

7. Click a link in the search result to view the respective employee record.

 Employee

Employee Details

EMP_NO: 10008
NAME: DAVID PELTON
DESIGNATION: DEVELOPER
SALARY: 12000
JOIN_DATE: 2006-05-18T18:30:00Z
ADDRESS: 7645 Down St, New York 98153

Summary

In this tutorial, you created a WCF service for the Oracle E-Business Suite artifacts you want to access from a SharePoint Portal. You also created an application definition for the Oracle E-Business Suite artifacts that is imported into a SharePoint portal to create Web Parts to present and search data in Oracle E-Business Suite.

See Also

Other Resources

[Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#)

Community Resources

There are two principal sources of information from Microsoft about the BizTalk Adapter Pack:

- The documentation and samples installed with the BizTalk Adapter Pack.
- The Microsoft BizTalk Adapter Pack forums.

BizTalk Adapter Pack Documentation

You can install the BizTalk Adapter Pack documentation when you install the BizTalk Adapter Pack. (For information about how to install the BizTalk Adapter Pack, see the installation guide. The installation guide is typically available at <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.) The core documentation covers the concepts and procedures required to use the BizTalk Adapter Pack.

After you install the BizTalk Adapter Pack Setup program, you can access the documentation through the **Microsoft BizTalk Adapter Pack** program group on the Start menu.

BizTalk Adapter Pack Samples

This BizTalk Adapter Pack release provides samples demonstrating how to use the adapters with:

- BizTalk Server.
- The Windows Communication Foundation (WCF) channel model.
- The WCF service model.
- Microsoft Office SharePoint Server.

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>.

BizTalk Adapter Pack Forums

You can post your queries about the BizTalk Adapter Pack on the forums.

Resource	Location
BizTalk Adapter Pack Forums	http://go.microsoft.com/fwlink/?LinkId=87695

See Also

Other Resources

[Getting Started](#)

Frequently Asked Questions

This section contains frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle E-Business Suite and the BizTalk Adapter Pack that you might have while learning or using the product. The FAQs are divided into four categories: General, BizTalk Server, WCF Service Model, and WCF Channel Model.

However, if you have specific questions related to Oracle E-Business adapter or BizTalk Adapter Pack that are not listed in this section, you can post your query on the BizTalk Adapter Pack forum at <http://go.microsoft.com/fwlink/?LinkId=87695>.

In This Section

- [General FAQs](#)
- [BizTalk Server FAQs](#)
- [WCF Service Model FAQs](#)
- [WCF Channel Model FAQs](#)

See Also

Other Resources

[Getting Started](#)

General FAQs

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle E-Business Suite and the BizTalk Adapter Pack.

What are the supported BizTalk versions for the BizTalk Adapter Pack?

The BizTalk Adapter Pack is supported with Microsoft BizTalk Server 2010.

What privileges are required to run the setup wizard?

You must always run the BizTalk Adapter Pack setup wizard with a user account that has administrative privileges on the computer.

What is WCF? Where can I read more about it?

WCF stands for Windows Communication Foundation. WCF is a programming framework developed by Microsoft for building service-oriented applications. WCF is a part of .NET framework, and enables developers to build secure, reliable, and transacted solutions that integrate across platforms and interoperate with existing investments. For more information about WCF, see <http://go.microsoft.com/fwlink/?LinkID=196850>.

What is WCF LOB Adapter SDK? Where can I read more about it?

The WCF LOB Adapter SDK is a collection of tools and components that provide a consistent framework for developing reusable, metadata-rich adapters for line-of-business systems. Adapters written using the WCF LOB Adapter SDK are surfaced as custom WCF bindings and can be consumed by a WCF-capable client.

For more information about WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation, which is installed along with the WCF LOB Adapter SDK at the following location on your computer: <installation drive>\Program Files\WCF LOB Adapter SDK\Documents.

How do I get started using the adapter?

To get started with using the Oracle E-Business adapter, you must figure out a role that suits you as a user of the adapter. The potential roles that are users of the Oracle E-Business adapter are the following: Developers, IT professionals, and Independent Software Vendors (ISVs). For information about the prerequisites for each of these roles, see [Prerequisite Skills and Knowledge](#). For a list of tasks that are applicable for various users of the Oracle E-Business adapter, see [Common Administrative Tasks](#) and [Common Developer Tasks](#).

How can I use the Oracle E-Business adapter to communicate with Oracle E-Business Suite?

You can use the Oracle E-Business adapter to communicate with Oracle E-Business Suite either by developing BizTalk applications, using the WCF service model or using the WCF channel model. For more information, see [Overview of BizTalk Adapter for Oracle E-Business Suite](#).

What interfaces are supported by the Oracle E-Business adapter for retrieving metadata?

The Oracle E-Business adapter supports two interfaces for retrieving metadata:

- MetadataExchange provided by WCF. WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from Oracle E-Business Suite.
- IMetadataRetrievalContract provided by the WCF LOB Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

How does the adapter connect to Oracle E-Business Suite?

The Oracle E-Business adapter uses Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7, Oracle Data Access Components for Oracle Client 11.1.0.7. The adapter clients must provide a connection string, [connection Uniform Resource Identifier \(URI\)](#), to connect to Oracle E-Business Suite. Internally, the adapter maps the connection URI to a connection string to connect to the underlying database in Oracle E-Business Suite. For more information, see [Establishing a Connection to the Oracle E-Business Suite](#).

Does the Oracle E-Business adapter provide a secure way of communicating with the Oracle E-Business Suite? Are there any best practices to ensure data security?

The Oracle E-Business adapter supports Enterprise Single Sign-On (SSO) for authentication on the connections that it

establishes with Oracle E-Business Suite. The SSO uses a database and a master secret to encrypt and store user credentials. It also provides services to map Microsoft Windows accounts to secondary credentials that are used to access a back-end system.

The Oracle E-Business adapter also requires you to enter username and password credentials to connect to Oracle E-Business Suite. These credentials are used to authenticate the user, and thus provide a level of authorization for the connections. The Oracle E-Business adapter provides a number of methods through which you can supply these credentials. For information about how to securely provide Oracle credentials in BizTalk solutions, see [Security Considerations When Using the Adapter with BizTalk Server](#). For information about how to securely provide Oracle credentials in programming solutions, see [Security Considerations When Programming on the Adapter](#).

For more information about:

- Data security in the Oracle E-Business adapter, see [Security and Protection](#).
- Best practices to ensure data security in the Oracle E-Business adapter, see [Best Practices](#).

Is there a GUI provided by the Oracle E-Business adapter to view and perform operations on the artifacts in Oracle E-Business Suite?

The Consume Adapter Service BizTalk Project Add-in and the Add Adapter Service Reference Visual Studio Plug-in provide a dialog box where you can view and perform operations on the artifacts in Oracle E-Business Suite. For more information about the GUI provided by the Oracle E-Business adapter, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

What are binding properties in the Oracle E-Business adapter? Where can I find information about all the binding properties in the Oracle E-Business adapter?

Adapter clients can use binding properties in the Oracle E-Business adapter to configure and control the adapter's behavior. For information about all the binding properties surfaced in the Oracle E-Business adapter, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

What are inbound and outbound operations in the Oracle E-Business adapter?

In inbound operations, the LOB system (Oracle E-Business Suite) is the client and the adapter clients are the service, wherein the transactions originate from Oracle E-Business Suite. For example, the Polling and Notification operations.

In outbound operations, the adapter clients are the client and the LOB system (Oracle E-Business Suite) becomes the service, wherein the transactions originate from the adapter clients. For example, the Insert, Select, Update, and Delete Operation on Interface tables, operation on stored procedures and functions, and composite operations.

Where can I find information about the Oracle data types that are supported in the Oracle E-Business adapter?

For information about the Oracle data types supported in the Oracle E-Business adapter, see [Basic Oracle Data Types](#).

What is applications context? How can I set applications context for various Oracle artifacts in the Oracle E-Business adapter?

For information about applications context and how to set it in the Oracle E-Business adapter, see [Setting Application Context](#).

Which approach (BizTalk Server, WCF service model or WCF channel model) can I use to perform various operations using the Oracle E-Business adapter?

For information about the approach you can use to perform various operations using the Oracle E-Business adapter, see [Development](#).

Does the Microsoft BizTalk Adapter Pack support tracing?

Microsoft BizTalk Adapter Pack enables adapter clients to activate Windows Communication Foundation (WCF) tracing and adapter-specific tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

Where are the trace files stored?

The trace files are stored at a location that the adapter clients specify. The adapter clients can specify this location when enabling tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

How do I view the traces?

You can use the WCF Service Trace Viewer tool to view the traces. For more information about the tool, see "Using Service Trace Viewer for Viewing Correlated Traces and Troubleshooting" at <http://go.microsoft.com/fwlink/?LinkId=91243>.

See Also

Other Resources

[Frequently Asked Questions](#)

BizTalk Server FAQs

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle E-Business Suite and the BizTalk Server.

Which BizTalk Server tools are used while working with SQL adapter? Where can I find more information about these tools?

You should have a working knowledge of the following BizTalk Server tools: BizTalk Explorer, Orchestration Designer, Pipeline Designer, BizTalk Mapper, and BizTalk Server Administration console. For more information about these tools, see [Learn About BizTalk Server Tools](#).

Can I reuse bindings of a BizTalk application? How?

A binding creates a mapping between a logical endpoint such as an orchestration port or a role link and a physical endpoint such as a send and receive port. This enables communication between different components of a BizTalk business solution. The binding information is stored in an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. You can export the bindings of a BizTalk assembly, application, or group, and then reuse it by importing into any other BizTalk application or group. For more information, see [Reusing Adapter Bindings](#).

What is the "Transaction Isolation Level"? How can I configure it?

The transaction isolation level determines the degree to which a transaction is isolated from the data changes made by other transactions. It defines the locking behavior of the Transact-SQL commands issued by a connection to the Oracle database. For more information about the transaction isolation level, see <http://go.microsoft.com/fwlink/?LinkId=126985>.

For instructions about configuring transaction isolation level in BizTalk Server, see [Configure Transaction Isolation Level and Transaction Timeout](#).

See Also

Other Resources

[Frequently Asked Questions](#)

WCF Service Model FAQs

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle E-Business Suite and the Windows Communication Foundation (WCF) service model.

What is the WCF service model?

The WCF service model is a programming model provided by WCF in which the LOB system (Oracle E-Business Suite) is exposed as a WCF service. The service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The WCF service model generates a proxy class—the WCF client class—through which your code can invoke operations and receive data using the Oracle E-Business adapter. For more information about the WCF service model, see [Developing Applications By Using the WCF Service Model](#).

Why would I want to use the WCF service model rather than the WCF channel model?

The WCF service model presents a model that is familiar to .NET programmers and that hides the underlying complexities of SOAP message exchange over a channel. Moreover, the Add Adapter Service Reference Plug-in is integrated with the Visual Studio design experience, and presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations exposed by the adapter. Therefore, the WCF service model is often the best choice to develop programming solutions for the Oracle E-Business adapter.

How do I get started with the WCF service model?

You can use either of the following tools provided by the WCF service model to generate a WCF client class or a WCF service contract and associated helper code from the service metadata that the Oracle E-Business adapter exposes:

- The ServiceModel Metadata Utility Tool (svcutil.exe), which ships with WCF.
- The Add Adapter Service Reference Visual Studio Plug-in, which ships with the Oracle E-Business adapter.

For more information about how to generate a WCF client or a WCF service contract, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

See Also

Other Resources

[Frequently Asked Questions](#)

WCF Channel Model FAQs

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Oracle E-Business Suite and the Windows Communication Foundation (WCF) channel model.

What is the WCF channel model?

The WCF channel model is a low-level abstraction of the SOAP message exchange between clients and services. It provides interfaces and types that enable you to send and receive messages by using a layered protocol stack called a channel stack. Each layer of the stack is composed of a channel, and each channel is created from a WCF binding. The Oracle E-Business adapter is a WCF custom transport binding that exposes an Oracle database as a WCF service. For more information about the WCF channel model, see [Overview of Using the WCF Channel Model with the Adapter](#).

Why would I want to use the WCF channel model rather than the WCF service model?

You would want to use the WCF channel model over the WCF service model because:

- The WCF channel model provides more fine-grained control over the operations that you perform on the Oracle database because in the WCF channel model, you directly control the contents of the messages that you send over the channel.
- The WCF channel model provided more comprehensive support for end-to-end streaming of Oracle large object (LOB) data types than the WCF service model. This is because in the WCF channel model, you directly control how you provide the message body on outgoing messages and how you process the message body on incoming messages.

How do I get started with the WCF channel model?

Using the WCF channel model, you can invoke operations and receive the results of a polling query by exchanging SOAP messages with the Oracle E-Business adapter over a WCF channel. To get started, you need to create outbound (client) and inbound (service) channels. For more information about creating channels in the WCF channel model, see [Creating a Channel](#).

See Also

Other Resources

[Frequently Asked Questions](#)

Planning and Architecture

This section describes the architecture for the Microsoft BizTalk Adapter for Oracle E-Business Suite. An understanding of the Oracle E-Business adapter architecture can help you:

- Understand the relationship between the Oracle E-Business adapter and the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
- Understand security boundaries, so that you can better secure data in your solution.
- Understand the Oracle E-Business adapter binding properties.
- Troubleshoot installation issues.

The following topics describe the architecture of end-to-end solutions that use the Oracle E-Business adapter to operate on an Oracle database and the internal architecture of the Oracle E-Business adapter.

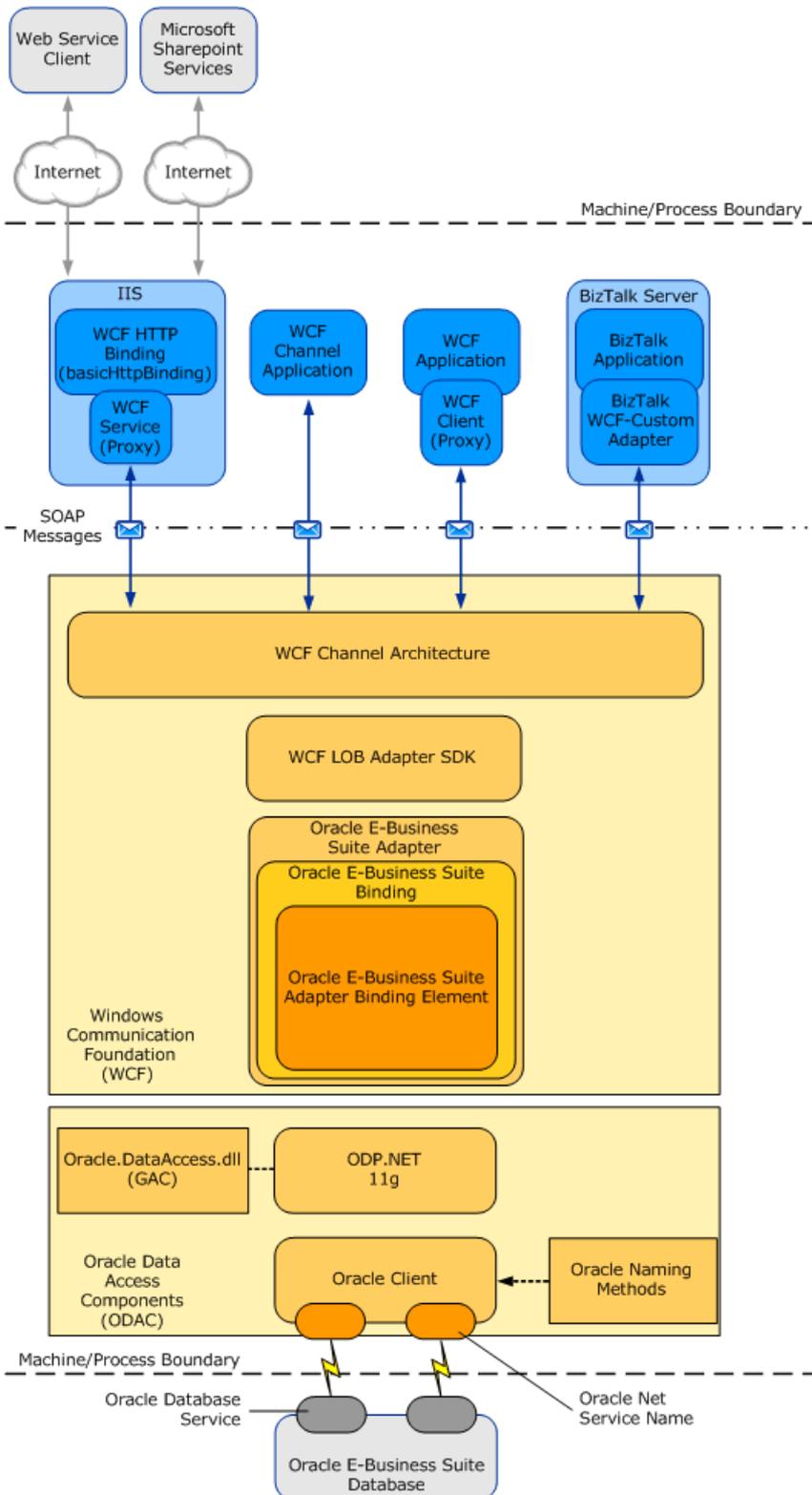
In This Section

- [Architecture of BizTalk Adapter for Oracle E-Business Suite](#)
- [Oracle E-Business Suite Adapter and the WCF LOB Adapter SDK](#)

Architecture of BizTalk Adapter for Oracle E-Business Suite

The Microsoft BizTalk Adapter for Oracle E-Business Suite is a Windows Communication Foundation (WCF) custom binding. This binding contains a single custom transport binding element that enables communication with an Oracle E-Business Suite. The Oracle E-Business adapter is wrapped by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK runtime and is exposed to applications through the WCF channel architecture. The Oracle E-Business adapter communicates with the Oracle E-Business Suite through the Oracle Data Provider for .NET (ODP.NET) and the Oracle client, which are part of the Oracle Data Access Components (ODAC) for Windows.

The following figure shows the end-to-end architecture for solutions that are developed by using the Oracle E-Business adapter.



Consuming the Adapter

The Oracle E-Business adapter exposes the Oracle E-Business Suite as a WCF service to client applications. To perform operations and access data on the Oracle E-Business Suite, client applications exchange SOAP messages with the Oracle E-Business adapter through WCF channels. The preceding figure shows four ways in which the Oracle E-Business adapter can be consumed.

- Through a WCF channel model application. A WCF channel model application performs operations on the Oracle E-Business Suite by using the WCF channel model to exchange SOAP messages directly with the Oracle E-Business adapter.
- Through a WCF service model application. A WCF service model application calls methods on a WCF client to perform operations on the Oracle E-Business Suite. A WCF client models the operations exposed by the Oracle E-Business adapter as .NET methods. You can use the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK or the WCF ServiceModel Metadata Utility Tool (svcutil.exe) to create a WCF client class from metadata exposed by the Oracle E-Business adapter.
- Through a BizTalk receive location or send port that is configured to use the Microsoft BizTalk WCF-Custom adapter. The WCF-Custom adapter enables the use of WCF extensibility features. By using the WCF-Custom adapter you can select and configure the Oracle EBS Binding and the behavior for the receive location or send port. For more information about how to use the Oracle E-Business adapter in BizTalk Server solutions, see [Developing BizTalk Applications](#).
- Through an IIS-hosted Web service. In this scenario, a WCF service proxy generated by using the adapter is hosted in IIS using the basicHttpBinding WCF binding. This exposes the service contract as a Web service to external users. IIS automatically hosts the adapter at runtime, which, in turn, communicates with the Oracle E-Business Suite.

The Oracle E-Business adapter and ODAC are always hosted in-process with the application or service that consumes the adapter.

The Oracle E-Business Adapter and WCF

WCF presents a programming model based on the exchange of SOAP messages over channels between clients and services. These messages are sent between endpoints exposed by a communicating client and service. An endpoint consists of:

- An *endpoint address*, which specifies the location at which messages are received
- A *binding*, which specifies the communication protocols used to exchange messages
- A *contract*, which specifies the operations and data types exposed by the endpoint.

A binding consists of one or more binding elements that stack on top of each other to define how messages are exchanged with the endpoint. At a minimum, a binding must specify the transport and encoding that are used to exchange messages with the endpoint. Message exchange between endpoints occurs over a channel stack that is composed of one or more channels. Each channel is a concrete implementation of one of the binding elements in the binding configured for the endpoint. For more information about WCF and the WCF programming model, see the WCF Help documentation at <http://go.microsoft.com/fwlink/?LinkID=196850>.

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes a WCF custom binding, the Oracle E-Business Suite Binding (**Microsoft.Adapters.OracleEBS.OracleEBSBinding**). By default, this binding contains a single custom transport binding element, the Oracle E-Business Suite Adapter Binding Element (**Microsoft.Adapters.OracleEBS.OracleEBSAdapter**), which enables operations on an Oracle E-Business Suite.

Microsoft.Adapters.OracleEBS.OracleEBSBinding (the Oracle E-Business Suite Binding) and **Microsoft.Adapters.OracleEBS.OracleEBSAdapter** (the Oracle E-Business Suite Adapter Binding Element) are public classes and are also exposed to the configuration system. Because the Oracle E-Business Suite Adapter Binding Element is exposed publicly, you can build your own custom WCF bindings capable of extending the functionality of the Oracle E-Business adapter. For example, you could implement a custom binding to support Enterprise Single Sign-on (SSO) in a WCF channel or service model solution. The reasons for doing this would be to aggregate database operations into a single multifunction operation or to perform schema transformation between operations implemented by a custom application and operations on the Oracle E-Business Suite.

The Oracle E-Business adapter is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and runs on top of the WCF LOB Adapter SDK runtime. The WCF LOB Adapter SDK provides a software framework and tooling infrastructure that the Oracle E-Business adapter uses to provide a rich set of features to users and

adapter clients.

ODP.NET

The Oracle E-Business adapter connects with the Oracle E-Business Suite through the ODP.NET and the Oracle client. Both of these components are part of Oracle Data Access Components (ODAC).

ODP.NET implements a data provider for the Oracle E-Business Suite that is consistent with the ADO.NET interface. The Oracle E-Business adapter uses the classes exposed by ODP.NET to operate on the Oracle E-Business Suite.

The Oracle client provides connectivity to the Oracle E-Business Suite. You establish a connection to an Oracle E-Business Suite by providing a connection URI to the Oracle E-Business adapter. You can specify the connection URI in two ways:

- **Using tnsnames.ora.** In this approach, the connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, port number, etc. from the net service name entry in the file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.
- **Without using tnsnames.ora.** In this approach, the adapter clients specify the connection parameters directly in the connection URI. This does not require the net service name to be present in the tnsnames.ora file on the client computer. This approach does not even require the tnsnames.ora file to be present on the client computer.

For more information about the connection URI, see [Establishing a Connection to the Oracle E-Business Suite](#). For more information about installing ODAC, see [Installing the BizTalk Adapter Pack](#).

See Also

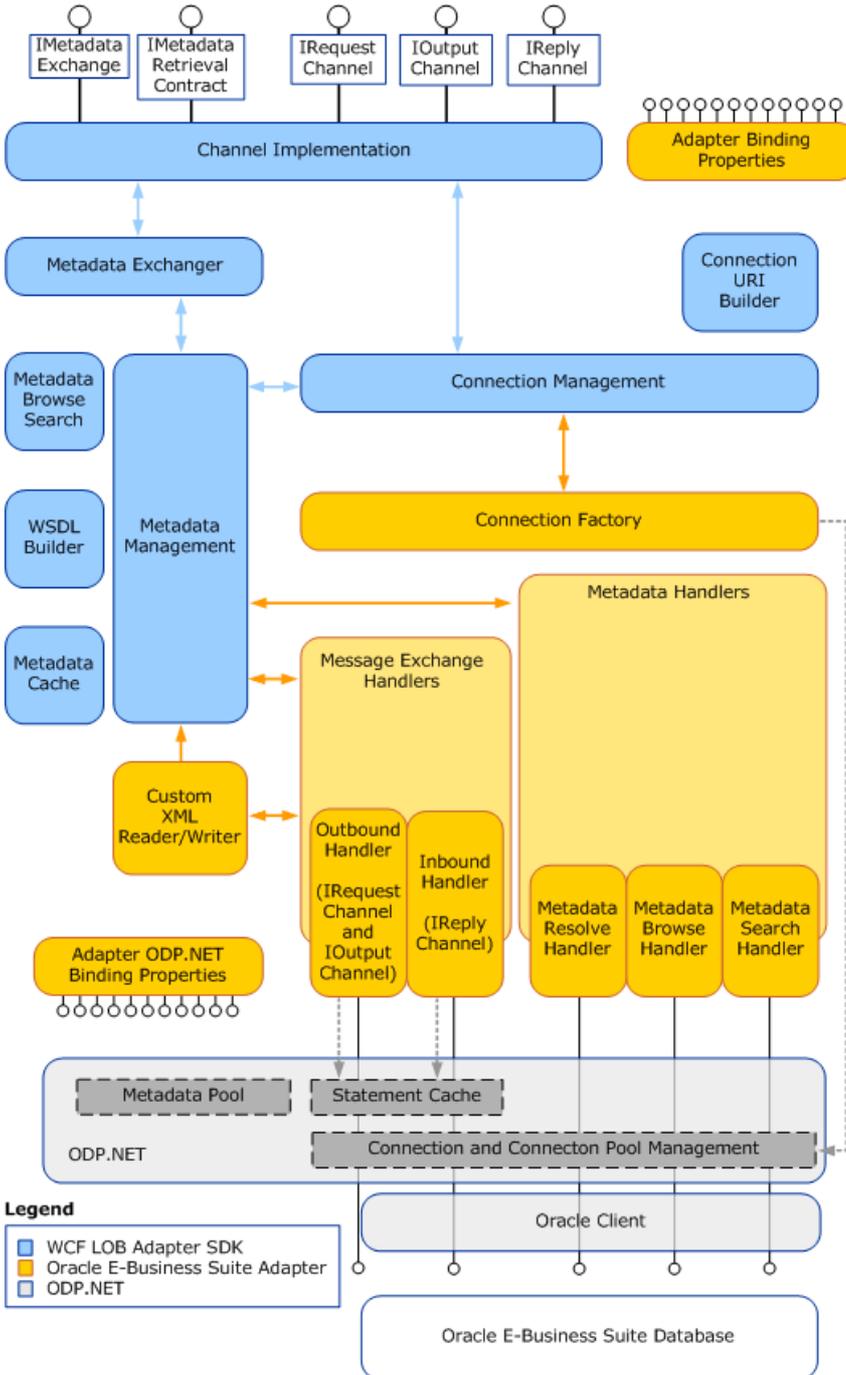
Other Resources

[Planning and Architecture](#)

Oracle E-Business Suite Adapter and the WCF LOB Adapter SDK

The Microsoft BizTalk Adapter for Oracle E-Business Suite implements a set of core components that uses functionality provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and provides connectivity to the Oracle database through Oracle Data Provider for .NET (ODP.NET).

The WCF LOB Adapter SDK serves as the software layer through which the Oracle E-Business adapter interfaces with the Windows Communication Foundation (WCF). ODP.NET serves as the layer through which the Oracle E-Business adapter interfaces with the Oracle database. The following figure shows the relationships between the internal components of the Oracle E-Business adapter, WCF LOB Adapter SDK, and ODP.NET.



See Also

Other Resources

[Planning and Architecture](#)

Security and Protection

Oracle E-Business applications deal with sensitive business information such as customer account details. Applications that use the Microsoft BizTalk Adapter for Oracle E-Business Suite to access and modify this information either locally or across a distributed network might inadvertently expose it to access by unauthorized actors, unless efforts are made to protect and secure the data during transmission. Data protection and security are usually thought of in the following terms:

- *Authorization* controls access to a resource based on the identity of the requester.
- *Authentication* provides mechanisms for verifying the identity of a requester.
- *Data confidentiality* provides mechanisms for protecting the privacy of data through encryption.
- *Data integrity* provides mechanisms to digitally sign data, so that the receiver can ensure that the data has not been altered in-transit.

Another important area of concern is the user-name password credentials that you supply to the Oracle E-Business adapter. The adapter uses these credentials to open connections to the Oracle database. These credentials can be supplied in the connection URI; however, because the user name and password are clear text, the Oracle E-Business adapter provides alternative methods that you can use to supply these credentials in a more secure manner.

The topics in this section provide guidelines to help you better secure the solutions that you develop with the Oracle E-Business adapter.

In This Section

- [Security Considerations Between Oracle E-Business Suite and the Adapter](#)
- [Security Considerations When Using the Adapter with BizTalk Server](#)
- [Security Considerations When Programming on the Adapter](#)
- [Best Practices](#)

Security Considerations Between Oracle E-Business Suite and the Adapter

The Oracle E-Business adapter provides no support for helping to secure communication between it and the Oracle database. You must provide a security mechanism to help ensure appropriate levels of authorization, authentication, data privacy, and data integrity for data exchanges between the adapter and the Oracle database.

One possible mechanism for helping to provide more security across the network is Internet Protocol Security (IPsec). IPsec is a framework of open standards for protecting communications over Internet Protocol (IP) networks. For more information about IPsec and about using IPsec with Microsoft products, see the Microsoft TechNet article "IPsec" at <http://go.microsoft.com/fwlink/?LinkID=196851>.

However, in the absence of security mechanisms like IPsec, the administrator must configure native Oracle data encryption and integrity to ensure secure data exchanges between the adapter client and Oracle E-Business Suite.

You must supply user name password credentials to the Oracle E-Business adapter. The Oracle E-Business adapter uses these credentials to authenticate the user on the Oracle database when it opens a connection. These credentials provide a level of authorization on the Oracle database for the connection.

Note

The credentials used by the Oracle E-Business adapter to establish a connection on the Oracle database do not provide message-level or transport-level authentication or authorization for data traveling across the network. They are only used to open a connection and authenticate the user on the Oracle database.

The Oracle E-Business adapter provides a number of methods through which you can supply these credentials. For information about how to more securely provide Oracle credentials in BizTalk solutions, see [Security Considerations When Using the Adapter with BizTalk Server](#). For information about how to more securely provide Oracle database credentials in programming solutions, see [Security Considerations When Programming on the Adapter](#).

Managing Audit Logs

Audit logs enable you to store information about the actions performed by various clients on your enterprise software, and helps usage monitoring and problem tracking. However, the Oracle E-Business adapter does not provide any way to manage audit logs for the actions performed by the adapter clients in Oracle E-Business Suite. This might pose a security threat as the adapter clients can repudiate the actions performed by them in Oracle E-Business Suite. To mitigate this issue, you must enable audit trail in Oracle to log the actions performed by the adapter clients in Oracle E-Business Suite. For information about how you can set up Oracle audit trails, see the "Oracle Workflow - Oracle E-Business Suite Process" white paper at <http://go.microsoft.com/fwlink/?LinkId=136388>.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Using the Adapter with BizTalk Server

When you configure a send port or a receive port (location) by using the BizTalk Server Administration console or use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas for a BizTalk solution, you must provide credentials for the Oracle E-Business Suite. It is important to supply these credentials in a secure way to help prevent them from being revealed to potentially malicious actors. This topic discusses how to most securely supply credentials for the Microsoft BizTalk Adapter for Oracle E-Business Suite for BizTalk Server solutions.

A more general discussion of security in the context of BizTalk solutions is an expansive topic and is beyond the scope of this documentation. For information about how you can make your BizTalk solutions more secure, see the "Security and Protection" topic in the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101135>.

How Do I Protect Credentials When I Use the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard?

When you use the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard to retrieve message schemas for a BizTalk solution, you can provide credentials to connect to Oracle E-Business Suite in the following two places:

- On the **Security** tab in the **Configure Adapter** dialog box. This ensures that your credentials will not be displayed in the **Configure a URI** field of the Consume Adapter Service Add-in dialog box, where anyone with access to your computer screen can read them.
- In the **OracleUserName** and **OraclePassword** binding properties on the **Binding Properties** tab in the **Configure Adapter** dialog box. For security reasons, the **OraclePassword** binding property is not available in the binding file (XML file) file generated as a result of using the Consume Adapter Service Add-in.

Note

Using the Add Adapter Metadata Wizard does not generate any binding file.

For more information about how to retrieve message schemas by using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

How Do I Protect Credentials When I Configure a Send Port or a Receive Location?

BizTalk solutions use the Microsoft BizTalk WCF-Custom or WCF-Oracle EBS adapter to consume WCF services. The Oracle E-Business adapter is a WCF binding that enables clients to consume the Oracle E-Business Suite as if it were a WCF service. BizTalk solutions consume the Oracle E-Business adapter through send ports and receive locations that are configured to use the WCF-Custom or WCF-OracleEBS adapter, which is, in turn, configured to use the Oracle E-Business adapter as its transport. For more information about how to configure send ports and receive ports (receive locations), including how to configure the WCF-Custom adapter, see [Configuring a Physical Port Binding Using a Port Binding File](#).

You configure the Oracle E-Business Suite credentials from the **Credentials** tab of the **WCF-Custom Transport Properties** dialog box for send ports or from the **Other** tab of the **WCF-Custom Transport Properties** dialog box for receive locations. Because the WCF-Custom or WCF-Oracle EBS adapter supports Enterprise Single Sign-On (SSO), you can choose to provide either a user name and password or an SSO affiliate application on either of these tabs. The following topics discuss both options.

User Name Password Credentials

You should only supply a user name and password from the **Credentials** tab (for send ports) or the **Other** tab (for receive locations) in the **WCF-Custom Transport Properties** dialog box. This ensures the following:

- Your credentials will not be displayed in the **Address (URI)** field of the dialog box. This prevents those who have access to your screen (or who have permissions that enable them to view the send port or receive location properties) from seeing your credentials.
- Your password will not be written to the binding file if you export the send port or receive port binding. This prevents anyone from with access to the file from viewing your password.

OraclePassword Binding Property

The value that you specify in the **OraclePassword** binding property for the send or receive port is available in clear text when you export bindings from an application in BizTalk Server. Therefore, after exporting a binding file from an application in BizTalk Server, you must manually remove the value of the **OraclePassword** binding property from the binding file, and then specify it again on each host where the exported binding will be used.

Enterprise Single Sign-On and SSO Affiliate Applications

You can configure the WCF-Custom adapter to use Enterprise Single Sign-on (SSO) to get the credentials for the Oracle E-Business Suite. SSO uses a database and a master secret to encrypt and store user credentials. It also provides services to map Microsoft Windows accounts to secondary credentials that are used to access a back-end system. By using SSO, you can map a Windows account to a user name and password on the Oracle database.

SSO uses *affiliate applications* and *SSO mappings* to map credentials to the back-end system. An affiliate application is a logical entity in SSO that refers to a system or an application that requires secondary credentials. An SSO mapping is associated with an affiliate application. It maps a Windows account to the secondary credentials used by that account to access the affiliate system or application. An SSO mapping can be associated with a Windows user account or with a group.

To use SSO with the Oracle Database adapter, you must do the following.

1. Create an affiliate application in SSO to hold the user name password credentials for the Oracle E-Business Suite. This step is often performed by someone with special types of SSO administrative privileges.
2. Create a user or group mapping for the affiliate application that maps your Windows account to the user name and password that are used to establish a connection with the Oracle database. Depending on your installation, a user might be able to perform this step or it might require someone with special types of SSO administrative privileges.

Note

When configured for SSO, the WCF-Custom adapter uses services provided by SSO to get the Oracle user name and password from the SSO database. It provides these (unencrypted) to the Oracle E-Business adapter, so that the adapter can open a connection to the Oracle E-Business Suite. SSO provides no encryption or protection across the connection between the Oracle E-Business adapter and the Oracle E-Business Suite.

For information about how to use SSO, including information about how to create affiliate applications and SSO mappings, see the "Using SSO" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103052>. For more general information about SSO, see the "Implementing Enterprise Single Sign-On" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103054>.

The AcceptCredentialsInUri Binding Property

The Oracle E-Business adapter surfaces the **AcceptCredentialsInUri** binding property. This property determines whether Oracle database or Oracle E-Business Suite credentials are permitted in the connection URI. By default, **AcceptCredentialsInUri** is **false** and the Oracle E-Business adapter throws an exception if credentials are included in the URI.

This property is surfaced because there are certain programming scenarios that require the credentials to be present in the connection URI. This should never be the case when you are configuring a send port or a receive location, or when you are using the Consume Adapter Service Add-in to retrieve message schemas from the Oracle E-Business adapter. It is recommended that you do not set **AcceptCredentialsInUri** to **true**. For more information about the Oracle E-Business adapter binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

The **AcceptCredentialsInUri** binding property is not available in BizTalk Server in the **Binding** tab while configuring a WCF-Custom or WCF-OracleEBS receive or send port. To set the value of the **AcceptCredentialsInUri** binding property, you must open the adapter bindings file (XML file) that is created after you have generated metadata using the Consume Adapter Service Add-in, and then locate this binding property in the file. Specify an appropriate value for this binding property, save the binding file, and then import the binding file in BizTalk Server. See [Importing Bindings](#) for instructions.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Programming on the Adapter

How Do I Protect Credentials When I Use the Add Adapter Service Reference Visual Studio Plug-in?

When you use the Add Adapter Service Reference Plug-in to create a WCF client, you must supply a user name and password for the Oracle E-Business Suite. You should only do this from the **Security** tab on the **Configure Adapter** dialog box. By entering the Oracle credentials from the **Security** tab instead of directly into the **Configure a URI** field, you ensure the following:

- The credentials will not be displayed in the **Configure a URI** field of the Add Adapter Service Reference Plug-in dialog box where anyone with access to your computer screen can read them.
- The credentials will not appear in the configuration file that the Add Adapter Service Reference Plug-in generates.

For more information about how to generate a WCF client by using the Add Adapter Service Reference Plug-in, including how to enter a user name and password for the Oracle E-Business Suite, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

What Are Best Practices for Setting Credentials in Code?

WCF provides the **ClientCredentials** class to help you configure the credentials that a client communication object, such as a **ChannelFactory**, uses to authenticate itself with a service. By using the **ClientCredentials** class, you ensure that WCF takes whatever authentication mechanisms are specified in that object's channel stack and applies them to the exchange between your client and the service.

Because the Microsoft BizTalk Adapter for Oracle E-Business Suite is hosted in-process with its consuming application, it is not imperative to use the **ClientCredentials** class to set credentials on the client communication objects that the consuming application uses. It is, however, considered good practice to do so.

The Oracle E-Business adapter encourages the use of the **ClientCredentials** class. This property specifies whether the adapter will accept the user name and password for the Oracle database in the connection URI. **AcceptCredentialsInUri** defaults to **false**, which means that the adapter will throw an exception if the connection URI contains credentials. You can set **AcceptCredentialsInUri** to **true** to supply credentials in the connection URI, if required in the client application.

The following example shows how to use the **Credentials** property to set credentials for the Oracle E-Business Suite on a **ChannelFactory**.

```
// Create binding and endpoint
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs-instance");

// Create the channel factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, address);

// Set user name and password
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open the channel factory
factory.Open();
```

The following example shows how to use the **ClientCredentials** class to set credentials for the Oracle E-Business Suite on a WCF client.

```
// Initialize a new client for the SQLEXECUTE operation from configuration
ConcurrentPrograms_ARClient client =
    new ConcurrentPrograms_ARClient("OracleEBSBinding_ConcurrentPrograms_AR");

// Set user name and password
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

```
// Open the client  
client.Open();
```

How Can I Provide for More Secure Data Exchange Across Process Boundaries?

The Oracle E-Business adapter is hosted in-process with the application or service that consumes it. Because the adapter is hosted in-process with the consumer, there is no need to provide security on messages exchanged between the consumer and the Oracle E-Business adapter. However, if the consuming application or service sends messages that contain sensitive database information across a process boundary to another service or client, you should take measures to provide adequate protection for this data in your environment. Windows Communication Foundation (WCF) provides many options for helping to secure messages sent between clients and services. For more information about helping to secure messages sent between clients and services in WCF, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>. For more general information about security features that WCF provides, see "Windows Communication Foundation Security" at <http://go.microsoft.com/fwlink/?LinkId=89726>.

See Also

Other Resources

[Security and Protection](#)

Best Practices

This section provides best practices that you should follow to more completely protect sensitive data when you use or develop applications that consume the Microsoft BizTalk Adapter for Oracle E-Business Suite.

Security Best Practices for the Connection between the Oracle E-Business Adapter and the Oracle Database

- The Oracle E-Business adapter provides no support for helping to secure communication between it and the Oracle E-Business Suite. You must provide a mechanism to help ensure an adequate level of security for data exchanged between the adapter and the Oracle database.
- Do not provide user name password credentials for the Oracle database in the connection URI. See the following sections for alternative methods of providing credentials to the Oracle E-Business adapter.
- The Oracle E-Business adapter also enables you to use Windows Authentication while connecting to Oracle E-Business Suite to generate metadata and perform operations, either through Visual Studio or BizTalk Server. Before using Windows Authentication, you must perform the steps listed in [Connecting to Oracle E-Business Suite Using Windows Authentication](#).

For more information, see [Security Considerations Between Oracle E-Business Suite and the Adapter](#).

Security Best Practices for Consuming the Oracle E-Business Adapter with BizTalk Server

- You should avoid providing user name password credentials for Oracle E-Business Suite in the connection URI.
- When you use the Consume Adapter Service Add-in, enter the user name password credential for Oracle E-Business Suite from the **Security** tab of the **Configure Adapter** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Oracle E-Business adapter on a send port, enter the user name password credential for the Oracle database from the **Credentials** tab of the **Configure WCF Custom Transport** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Oracle E-Business adapter on a receive location, enter the user name password credential for the Oracle database from the **Other** tab of the **Configure WCF Custom Transport** dialog box.
- After exporting a binding file from an application in BizTalk Server, you must manually remove the value of the **OraclePassword** binding property (available in clear text) from the binding file, and then specify it again on each host where the exported binding will be used.
- The Oracle E-Business adapter also enables you to use Windows Authentication while connecting to Oracle E-Business Suite to generate metadata and perform operations, either through Visual Studio or BizTalk Server. Before using Windows Authentication, you must perform the steps listed in [Connecting to Oracle E-Business Suite Using Windows Authentication](#).
- If an application that consumes the Oracle E-Business adapter sends messages that contain sensitive database information across a process boundary to another service or client, ensure that these messages have sufficient security measures applied to provide adequate data protection in your environment.

For more information, see [Security Considerations When Using the Adapter with BizTalk Server](#).

Security Best Practices for Consuming the Oracle E-Business Adapter with Programming Solutions

- You should avoid providing user name password credentials for the Oracle database in the connection URI.
- When you use the Add Adapter Service Reference Visual Studio Plug-in, enter the user name password credential for the Oracle database from the **Security** tab of the **Configure Adapter** dialog box.

- In WCF channel model programming, use the **Credentials** property on the channel factory to set the user name password credential for the Oracle database.
- In WCF service model programming, use the **ClientCredentials** property on the WCF client to set the user name password credential for the Oracle database.
- If an application that consumes the Oracle E-Business adapter sends messages that contain sensitive database information across a process boundary to another service or client, ensure that these messages have sufficient security measures applied to provide adequate data protection in your environment.
- The Oracle E-Business adapter also enables you to use Windows Authentication while connecting to Oracle E-Business Suite to generate metadata and perform operations, either through Visual Studio or BizTalk Server. Before using Windows Authentication, you must perform the steps listed in [Connecting to Oracle E-Business Suite Using Windows Authentication](#).

For more information see, [Security Considerations When Programming on the Adapter](#).

Security Best Practices for Hosting the Oracle E-Business Adapter in IIS

Hosting the Oracle E-Business adapter in Microsoft Internet Information Services (IIS) as a Web service exposes operations surfaced by the Oracle E-Business adapter to Web clients. These operations might involve exchanging sensitive data over the Internet, so you should take measures to help ensure that this data is as secure as possible.

WCF provides two standard bindings for HTTP transport: the **BasicHttpBinding** provides basic HTTP transport with no security mechanisms; the **WSHttpBinding** supports both transport-level and message-level security mechanisms.

You can either use the **BasicHttpBinding** over an HTTPS connection, or use the **WSHttpBinding** to help protect your data. The WCF LOB Adapter SDK includes the WCF LOB Adapter Service Development Wizard to generate WCF service for LOB artifacts. This wizard only supports use of **BasicHttpBinding**.

You can also develop a custom HTTP binding to leverage additional security mechanisms that your environment provides. For more information about the security features that WCF provides, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>.

Security Best Practices for WCF Diagnostic Tracing and Message Logging

WCF supports diagnostic tracing and message logging. You configure diagnostic tracing and message logging either through configuration files or by using Windows Management Instrumentation (WMI). Depending on the configuration options you set, WCF diagnostic tracing or message logging can emit sensitive information to log files, where it could potentially be exposed to observation by unauthorized users.

Follow the recommendations provided in the WCF documentation to mitigate potential security threats exposed by enabling these features. At a minimum, you should observe the following best practices for diagnostic tracing and message logging:

- Do not enable "verbose" or "information" tracing in a production environment. This may lead to performance degradation. However, you must enable "warning" and "error" tracing in a production environment. If you enable tracing, you must take proper security measures to protect your data. See the WCF documentation for more information.
- Ensure that log files and configuration files are protected by access control lists (ACLs).

The following warnings apply specifically to the messages that are exchanged between a client application and the Oracle E-Business adapter:

- WCF diagnostic tracing can log the header (but not the body) of messages exchanged with the Oracle E-Business adapter. Because the message action is in the message header, this reveals the operations invoked on the Oracle E-Business adapter by the client.
- If WCF message logging is enabled and **logMessagesAtServiceLevel** is **true**, the message header (but not the message body) of messages exchanged between the adapter client and the Oracle E-Business adapter are logged. Because the message action is in the message header, this reveals the operations that the client invoked on the Oracle E-Business adapter. If **logEntireMessage** is also **true**, the message body will be logged. This can reveal sensitive database

information.

For more information about improving security when you enable diagnostic tracing, see "Security Concerns and Useful Tips for Tracing" at <http://go.microsoft.com/fwlink/?LinkId=89796>. For more information about improving security when you enable message logging, see "Security Concerns for Message Logging" at <http://go.microsoft.com/fwlink/?LinkId=89797>.

See Also

Other Resources

[Security and Protection](#)

Development

The Microsoft BizTalk Adapter for Oracle E-Business Suite is a Windows Communication Foundation (WCF) binding. Client applications can consume the Oracle E-Business adapter to invoke operations on Oracle E-Business Suite artifacts. The Oracle E-Business adapter can be consumed:

- Through a physical port binding in a BizTalk Server solution.
- By invoking methods on an instance of a client proxy.
- By sending SOAP messages over a channel instance in code that uses the WCF channel model.

The following table:

- Lists the different operations that can be performed on Oracle E-Business Suite using the Oracle E-Business adapter.
- Provides links to the topics containing information about performing the task using the chosen approach (BizTalk Server, WCF service model or WCF channel model).

O p e r a t i o n	BizTalk Server	WCF Service Model
P e r f o r m i n g o p e r a t i o n s o n i n t e r f a c e t a b e s a n d v i e w s	Performing Insert, Update, Delete, or Select Operations on Interface Tables and Interface Views Using BizTalk Server	Performing Insert, Update, Delete, or Select Operations on Interface Tables and Views Us

<p>Performing Operations on Tables with Large Data Types Using BizTalk Server</p>	<p>Performing Operations on Tables with Large Data Types Using the WCF Service Model</p>
<p>Performing Composite Operations on Oracle Database by Using BizTalk Server</p>	

<p>Invoking Concurrent Programs in Oracle E-Business Suite Using BizTalk Server</p>	<p>Invoking Concurrent Programs in Oracle E-Business Suite Using the WCF Service Model</p>
<p>Invoking Request Sets in Oracle E-Business Suite Using BizTalk Server</p>	<p>Invoking Request Sets in Oracle E-Business Suite Using the WCF Service Model</p>

<p>Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server</p>	<p>Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using WCF</p>
<p>Polling Oracle E-Business Suite Using BizTalk Server</p>	<p>Polling Oracle E-Business Suite Using the WCF Service Model</p>

R e c e i v i n g d a t a b a s e c h a n g e m e n t s o t i f i c a t i o n s f r o m O r a c l e E- B u s i n e s s S u i t e	Receiving Database Change Notifications Using BizTalk Server	Receiving Database Change Notifications Using the WCF Service Model
---	--	---

The topics in this section provide information, procedures, and examples to help you develop applications that consume the Oracle E-Business adapter in both BizTalk Server and .NET programming solutions. The topics also provide information on other key aspects of using the adapters such as:

- [Connecting to the Oracle E-Business Suite.](#)
- [Retrieving metadata from the Oracle E-Business Suite.](#)
- [Using binding properties to configure the adapter.](#)

In This Section

- [Establishing a Connection to the Oracle E-Business Suite](#)
- [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#)
- [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#)
- [Receiving Polling-based Data-changed Messages](#)
- [Before Developing Applications Using the Oracle E-Business Suite Adapter](#)
- [Developing BizTalk Applications](#)
- [Developing Applications By Using the WCF Service Model](#)
- [Developing Applications by Using the WCF Channel Model](#)
- [Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server](#)
- [Samples](#)
- [How Do I?](#)

Establishing a Connection to the Oracle E-Business Suite

The Microsoft BizTalk Adapter for Oracle E-Business Suite is a Windows Communication Foundation (WCF) custom binding. As such, it enables communication to Oracle E-Business Suite through a WCF endpoint address. In WCF the endpoint address identifies the network location of a service and is typically expressed as a Uniform Resource Identifier (URI). The Oracle E-Business adapter expresses this location as a connection URI, which contains properties that the Oracle E-Business adapter uses to establish a connection to Oracle E-Business Suite. You must specify a connection URI when you:

- Create a channel factory or a channel listener using the WCF channel model or when you create a WCF client or service host using the WCF service model.
- Create a physical port binding in a BizTalk Server solution.
- Use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class or WCF service interface for a WCF service model solution.
- Use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas from the Oracle E-Business adapter for a BizTalk Server solution.
- Use the ServiceModel Metadata Utility tool (svcutil.exe) to generate a WCF client class or WCF service interface for a WCF service model solution.

The Oracle E-Business adapter supports two ways of establishing a connection to the underlying Oracle database:

- **Using tnsnames.ora.** In this approach, the connection URI provided by the adapter client contains only the net service name specified in the tnsnames.ora file. The adapter extracts the connection parameters such as server name, service name, port number, etc. from the net service name entry in the file. To use this approach, the computer running the Oracle client must be configured to include the net service name for the Oracle database in the tnsnames.ora file.

◆ Important

Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the [connection URI](#) cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

- **Without using tnsnames.ora.** In this approach, the adapter clients specify the connection parameters directly in the connection URI. This does not require the net service name to be present in the tnsnames.ora file on the client computer. This approach does not even require the tnsnames.ora file to be present on the client computer.

◆ Important

This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

The topics in this section describe how to establish a connection between the Oracle E-Business adapter and Oracle E-Business Suite by providing you with:

- Information about configuring the Oracle client (required only if using the tnsnames.ora to establish the connection.)
- Information about the connection properties and the structure of the Oracle E-Business Suite connection URI.
- Links to topics that show how to specify a connection URI by using the Oracle E-Business adapter.
- Information about connecting to Oracle E-Business Suite using Windows Authentication.

In This Section

- [Configuring the Oracle Client](#)
- [The Oracle E-Business Suite Connection URI](#)
- [Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Configuring the Oracle Client

◆ Important

This topic is relevant only if you are using tnsnames.ora to connect to the Oracle database.

To establish a connection to the Microsoft BizTalk Adapter for Oracle E-Business Suite, the adapter connects to the underlying Oracle database through the Oracle client installed on your computer. The Oracle E-Business adapter passes the net service name that you specify in the connection URI to the Oracle client to establish a connection to Oracle E-Business Suite. The net service name is an alias that the Oracle client uses to acquire connection information for the target Oracle database service.

The Oracle client resolves the net service name according to the naming method that it is configured to use. You use the Oracle Net Configuration Assistant to configure the naming methods to be used by the Oracle client. The Oracle E-Business adapter supports the Local Naming method for connecting to Oracle E-Business Suite. This method uses a local file, tnsnames.ora, to resolve the net service name.

The tnsnames.ora file associates net service names with connect descriptors that contain the information that the Oracle client needs to establish a connection to a specific Oracle database service (instance). The following is a sample entry from tnsnames.ora.

```
ADAPTER =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = yourOracleServer)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = yourOracleDatabaseServiceName)
    )
  )
```

In this sample entry, ADAPTER is the net service name. The connect descriptor specifies address information and the service name of the Oracle database service associated with ADAPTER. You can use the Oracle Net Configuration Assistant to create and configure net service names in tnsnames.ora. After you have configured the net service name, you can specify it in a connection URI as in the following example.

```
oracleEBS://ADAPTER
```

For more information about using the Oracle Net Configuration Assistant and about tnsnames.ora, see the Oracle Database Net Services Administrator's Guide. Consult your database administrator about configuration details for your specific installation.

See Also

Other Resources

[Establishing a Connection to the Oracle E-Business Suite](#)

The Oracle E-Business Suite Connection URI

The Microsoft BizTalk Adapter for Oracle E-Business Suite connection URI contains properties that the adapter uses to establish a connection to Oracle E-Business Suite, and essentially the underlying Oracle database. The Oracle E-Business adapter supports two ways of connecting to the underlying Oracle database: using `tnsnames.ora` and without using `tnsnames.ora`. Based on the kind of connectivity approach, the format of the connection URI is also different. This topic provides information about the Oracle connection URI and also provides links to other topics that explain how to specify a URI in different programming scenarios.

Oracle E-Business Suite is an application layer that interfaces with an underlying Oracle database, and is categorized into different applications such as Financials and HR, based on the different needs within an organization. Each of these applications provides various “forms” that enable users to enter data into the underlying Oracle database. Access to these forms is restricted by associating users with an application context that comprises the organization ID to which a user belongs, the “responsibility” associated with the user, and the name of the Oracle E-Business Suite application that the user wants to invoke. Even though the adapter connects directly to the underlying database and does not use forms to interface with the Oracle E-Business Suite, setting the application context is mandatory when performing operations on the Oracle E-Business Suite artifacts. So, to connect to the Oracle E-Business suite, and the underlying Oracle database, using the Oracle E-Business adapter, you must:

- Specify a connection URI to connect to Oracle E-Business Suite and the underlying Oracle database. While establishing a connection, you can choose to specify the credentials for Oracle E-Business Suite or the underlying Oracle database.
- Set the application context for the user. The Oracle E-Business adapter exposes certain binding properties that accept the credentials and the responsibility. For more information about these binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For more information about setting the application context, see [Setting Application Context](#).

This section provides information about how to specify the connection URI to connect to the underlying database using `tnsnames.ora` and without using `tnsnames.ora`. It also provides information about using the connection URI to connect to Oracle E-Business Suite.

The Connection URI to Connect to the Underlying Oracle Database Using `tnsnames.ora`

Important

- For this approach, you must add the net service name entry in the `tnsnames.ora` file on the computer with the adapter client installed. For more information about the net service name entry, see [Configuring the Oracle Client](#).
- Due to an Oracle Client limitation, the **DataSourceName** parameter (net service name) in the connection URI cannot contain more than 39 characters if you are performing operations in a transaction. Therefore, make sure that the value specified for the **DataSourceName** parameter is less than or equal to 39 characters if you will be performing operations in a transaction.

The connection URI can contain an Oracle net service name that is used to identify the Oracle E-Business Suite service with which you want to connect. The Oracle client resolves the Oracle net service name that you provide in the connection URI to connection information for an Oracle E-Business Suite service, according to a hierarchy of Oracle naming methods that you configure it to use. One common naming method is called local naming. In local naming, the Oracle client uses a file called `tnsnames.ora` to resolve the Oracle net service name.

A typical endpoint address URI in WCF is represented as: `scheme://userauthparams@hostinfoparams`, where:

- `scheme` is the scheme name.
- `userauthparams` is a name-value collection of parameters required for user authentication by the endpoint.
- `hostinfoparams` is information required to establish the connection to the host; for example, a net service name.

The Oracle E-Business adapter connection URI adheres to this basic format and is implemented as follows:

```
oracleebcs://User=[USER_NAME];Password=[PASSWORD]@[NET_SERVICE_NAME]
```

The following table explains the properties contained in the connection URI.

Connection URI Property	Cat	Description
[USER_NAME]	use	<p>The user name to use for authentication. The Oracle E-Business adapter exposes a ClientCredentialType binding property that specifies the type of Oracle client credential that the client specifies to establish a connection. The possible values for the ClientCredentialType binding property are Database and EBusiness. Depending on the value of this binding property, you must specify relevant credentials. For more information, see Oracle Credentials and the Connection URI.</p> <p> Note You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI.</p> <p> Note The Oracle E-Business adapter does not preserve the case of the value that you enter for the user name when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.</p>
[PASSWORD]	use	<p>The password to use for authentication. The Oracle E-Business adapter exposes a ClientCredentialType binding property that specifies the type of Oracle client credential that the client specifies to establish a connection. If the ClientCredentialType property is set to Database, the clients must specify the password for an Oracle database user. If the ClientCredentialType property is set to EBusiness, the clients must specify the password for an Oracle E-Business Suite user.</p> <p> Note The Oracle E-Business adapter does not preserve the case of the value that you enter for the password when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.</p>
[NET_SERVICE_NAME]	hostinfo	<p>A net service name that is specified in the tnsnames.ora file on the computer where the Oracle E-Business adapter is installed. For more information about net service names and tnsnames.ora, see Configuring the Oracle Client.</p>

The Connection URI to Connect to the Underlying Oracle Database Without Using tnsnames.ora

 Important
<ul style="list-style-type: none"> For this approach, you need not have the net service name entry in the tnsnames.ora. Also, you do not even need to have the tnsnames.ora file on the computer with the adapter client installed. This mode of connectivity is not supported if you are performing operations in a transaction. This is due to a limitation of Oracle Client.

A typical endpoint address URI in WCF is represented as: `scheme://userauthparams@hostinfoparams`, where:

- `scheme` is the scheme name.
- `userauthparams` is a name-value collection of parameters required for user authentication by the endpoint.

- hostinfoparams is information required to establish the connection to the host; for example, server name, port number, etc.

The Oracle E-Business adapter connection URI adheres to this basic format and is implemented as follows:

```
oracleEBS://User=[USER_NAME];Password=[PASSWORD]@[SERVER_NAME]:[PORT_NUMBER]/[SERVICE_NAME]
/[SERVICE_TYPE]
```

The following table explains the properties contained in the connection URI.

Connection URI Property	Category	Description
[USER_NAME]	userparams	<p>The user name to use for authentication. The Oracle E-Business adapter exposes a ClientCredentialType binding property that specifies the type of Oracle client credential that the client specifies to establish a connection. The possible values for the ClientCredentialType binding property are Database and EBusiness. Depending on the value for this binding property, you must specify relevant credentials. For more information, see Oracle Credentials and the Connection URI.</p> <p> Note You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI.</p> <p> Note The Oracle E-Business adapter does not preserve the case of the value that you enter for the user name when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.</p>
[PASSWORD]	userparams	<p>The password to use for authentication. The Oracle E-Business adapter exposes a ClientCredentialType binding property that specifies the type of Oracle client credential that the client specifies to establish a connection. If the ClientCredentialType property is set to Database, the clients must specify the password for an Oracle database user. If the ClientCredentialType property is set to EBusiness, the clients must specify the password for an Oracle E-Business Suite user.</p> <p> Note The Oracle E-Business adapter does not preserve the case of the value that you enter for the password when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.</p>
[SERVER_NAME]	hostinfoparams	Name of the server on which the Oracle E-Business Suite is running. This is mandatory.
[PORT_NUMBER]	hostinfoparams	The Oracle Net Listener port. The default value 1521.

[SERVICE_NAME]	hostname	The Oracle database service name. This is mandatory.
[SERVICE_TYPE]	hostname	The type of Oracle service. The possible values are Dedicated or Shared . A dedicated service uses a dedicated server process to serve only one user process. A shared service uses a shared server process that can serve multiple user processes. Default is Dedicated .

Oracle Credentials and the Connection URI

By default, the Oracle E-Business adapter throws an exception when the Oracle credentials are specified in the connection URI. This is because these credentials are represented as plain text in the connection URI, and this poses a security risk. You can set the **AcceptCredentialsInUri** binding property to control whether the connection URI can contain credentials for the Oracle database. If the **AcceptCredentialsInUri** property is **false**, which is the default, the Oracle E-Business adapter throws an exception if the connection URI contains Oracle credentials; if the property is **true**, no exception is thrown.

Important

Due to the security risks posed by passing credentials in strings as plain text, you should avoid specifying Oracle database connection credentials in the connection URI. For more information about how to more securely provide credentials for the Oracle database, see [Security and Protection](#).

You may also choose to specify either the database credentials or the Oracle E-Business Suite credentials for establishing a connection to Oracle E-Business Suite. The adapter exposes three binding properties to enable this behavior:

ClientCredentialType, OracleUserName, OraclePassword.

The possible values for the **ClientCredentialType** binding property are **Database** and **EBusiness**.

- If the **ClientCredentialType** property is set to **Database**, the clients must specify the database credentials.
- If the **ClientCredentialType** property is set to **EBusiness**, the clients must specify the Oracle E-Business Suite credentials. In this case, the adapter clients must also specify the database credentials for the **OracleUserName** and **OraclePassword** binding properties.

Important

In scenarios where the adapter clients specify the database credentials to connect to Oracle E-Business Suite by setting the **ClientCredentialType** binding property to **Database**, but invoke an Oracle E-Business Suite artifact, the values specified for **OracleUserName** and **OraclePassword** binding properties are used for setting the application context. Setting the application context is mandatory for invoking artifacts in Oracle E-Business Suite. For more information about setting the application context, see [Setting Application Context](#).

Using Reserved Characters in the Connection URI

The Oracle E-Business adapter does not support specifying a connection URI that has special characters for any of the parameter values. If the connection parameter values contain special characters, make sure you do one of the following:

- If you are specifying the URI in Visual Studio using Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. If you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.
- If you are specifying the URI while creating a send or receive port in BizTalk Server Administration console, and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

Using the Connection URI to Connect to Oracle E-Business Suite

The following is an example of a connection URI for Oracle E-Business adapter using tnsnames.ora.

```
oracleebs://ADAPTER
```

In this example, ADAPTER is a net service name that is associated with the SERVICE NAME and connection information for the target Oracle database in tnsnames.ora.

The following is an example of a connection URI for Oracle E-Business adapter without using tnsnames.ora.

```
oracleebs://yourOracleServer:1521/yourOracleDatabaseServiceName/Dedicated
```

In this example, the server name is "yourOracleServer" and the service name is "yourOracleDatabaseServiceName".

For information about how to establish a connection to Oracle E-Business Suite when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).
- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).

See Also

Concepts

[Configuring the Oracle Client](#)

Other Resources

[Establishing a Connection to the Oracle E-Business Suite](#)

Connecting to Oracle E-Business Suite Using Windows Authentication

The Oracle E-Business adapter enables adapter clients to use Windows Authentication to establish a connection with the Oracle E-Business Suite. To use Windows Authentication adapter clients must specify a "/" for user name and leave the password blank. For more information about connecting to the Oracle E-Business Suite using Windows Authentication, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

To use Windows Authentication, you must do the following:

- If the **ClientCredentialType** property is set to **Database**, specify "/" for the user name and leave the password blank to connect to the Oracle E-Business Suite.
- If the **ClientCredentialType** property is set to **EBusiness**, specify the Oracle E-Business Suite credentials to connect. Also, you must specify "/" for the **OracleUserName** binding property and leave the **OraclePassword** binding property blank.

To enable adapter clients to use Windows Authentication to connect to an Oracle database, you must perform the following tasks on the computer running the Oracle database.

1. Make sure that the `sqlnet.ora` file on both the client and the server, available under `ORACLE_BASE\ORACLE_HOME\network\admin\sqlnet.ora`, has the following entry:

```
SQLNET.AUTHENTICATION_SERVICES= (NTS)
```

2. Connect to the Oracle database as SYSDBA.
3. Create the Windows user as an external user in Oracle database. Note that the user name must be in upper case.

```
CREATE USER "OPS$<DOMAIN_NAME>\<USER_NAME>" IDENTIFIED EXTERNALLY;
```

4. Grant privileges to the user.

```
GRANT CONNECT, RESOURCE TO "OPS$<DOMAIN_NAME>\<USER_NAME>";
```

5. The Oracle E-Business Suite artifacts are available under the APPS schema. To enable the newly created user, logging in using Windows Authentication, to access the Oracle E-Business Suite artifacts, the user's schema must be changed to the APPS schema. You can add the following SQL command to the logon script that changes the user's default schema to APPS when the user logs on.

```
alter session set current_schema=APPS;
```

6. Even though you changed the schema of the user to APPS schema, you will still not be able to see Oracle E-Business Suite artifacts while browsing and generating metadata using the Oracle E-Business adapter. This is because the newly created user does not have permissions for the APPS schema. Make sure you provided permission for the APPS schema for the newly created user.

See Also

Concepts

[Configuring the Oracle Client](#)

Other Resources

[Establishing a Connection to the Oracle E-Business Suite](#)

Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio

The Microsoft BizTalk Adapter for Oracle E-Business Suite provides three Visual Studio components that you can use to help you develop solutions using the adapter:

- Consume Adapter Service BizTalk Project Add-in
- Add Adapter Metadata Wizard
- Add Adapter Service Reference Visual Studio Plug-in

Adapter clients must use these components to connect to Oracle E-Business Suite, and then generate metadata for the operations they want to perform. All these Visual Studio components simplify development by:

- Providing a Microsoft Windows interface through which you can browse and search for operations that you want to use in your solution.
- Retrieving metadata exposed by the adapter for these target operations.
- Converting that metadata, which is expressed as a Web Services Description Language (WSDL) document by the adapter, into a form that you can use in your solution (XSD message schemas for BizTalk projects or a .NET object representation of a service contract for the WCF service model) and adding it to your project.

This section provides instructions about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

Note

If you have created a solution using the adapter on a particular version of Oracle E-Business Suite, and now want to deploy the solution on a different version of Oracle E-Business Suite then you should test the solution before deploying it. You might face issues while deploying the solution on a different version of Oracle E-Business Suite because the metadata of the underlying artifact might be different. To resolve this issue, you should regenerate the metadata using the adapter on the same version of Oracle E-Business Suite on which you intend to deploy the solution.

In This Section

- [Connecting to the Oracle E-Business Suite in Visual Studio](#)
- [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Connecting to the Oracle E-Business Suite in Visual Studio

This section provides information about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

- The **Consume Adapter Service BizTalk Project Add-in** is available in BizTalk Server projects. You use the Consume Adapter Service Add-in to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The **Add Adapter Metadata Wizard** is available in BizTalk Server projects. You use the Add Adapter Metadata Wizard to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).

Note

Because the Oracle E-Business adapter is exposed both as a WCF-Custom binding and as BizTalk adapter, you can use either the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard from a BizTalk project to connect to Oracle E-Business Suite.

- The **Add Adapter Service Reference Visual Studio Plug-in** is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications By Using the WCF Service Model](#).

To use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in, you must first connect to the Oracle E-Business Suite. All three methods present a dialog box through which you configure a connection by setting the following:

- **Connection parameters.** These are the parameters that are used to build the connection URI. You must specify a data source (Oracle net service name).
- **User name password credentials for the Oracle E-Business Suite.** These are used to authenticate you on the Oracle E-Business Suite when the connection is established. You must specify a user name and password.

Important

At this stage, you can specify the credentials for the Oracle E-Business Suite or the underlying Oracle database. To connect and generate metadata you can specify any credentials. However, while performing operation to invoke an Oracle E-Business Suite artifact, you must specify the Oracle E-Business Suite credentials because they are required to set the application context for the Oracle E-Business Suite application you want to invoke. For more information about setting applications context, see [Setting Application Context](#).

- **Binding properties.** Binding properties are optional at design-time, that is, while generating metadata for operations. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

At a minimum, when you configure the connection to the Oracle database, you only have to specify binding properties and connection parameters that are needed to establish the connection and that affect the metadata returned by the Oracle Database adapter for the operations you want to target. However, you might also want to specify values for any additional binding properties and connection parameters that will be used at run time. This is because:

- The Consume Adapter Service Add-in creates a BizTalk port binding file from the binding properties and connection parameters that you specify when you configure the connection, and adds this file to your project. Later, you can use this binding file to create a port in the BizTalk Server Administration console. For more information about the binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).
- The Add Adapter Service Reference Plug-in creates an app.config file from the binding properties and connection

properties that you specify when you configure the connection, and adds this file in your project directory.

In This Section

- [Connecting to Oracle E-Business Suite in Visual Studio Using Consume Adapter Service Add-in](#)
- [Connecting to Oracle E-Business Suite in Visual Studio Using Add Adapter Metadata Wizard](#)
- [Connecting to Oracle E-Business Suite in Visual Studio Using Add Adapter Service Reference Plug-in](#)

See Also

Other Resources

[Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#)

Connecting to Oracle E-Business Suite in Visual Studio Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in is installed when you install WCF LOB Adapter SDK. The Consume Adapter Service Add-in loads all the WCF-Custom bindings installed on the computer. To connect to Oracle E-Business Suite using the WCF-based Oracle E-Business adapter in a BizTalk project, you must use the **oracleEBSBinding**.

This topic provides instructions on how to use the Consume Adapter Service Add-in.

Connecting to Oracle E-Business Suite Using the Consume Adapter Service Add-in

Perform the following steps to connect to Oracle E-Business Suite using the Consume Adapter Service Add-in.

To connect to Oracle E-Business Suite

1. To connect using the Consume Adapter Service Add-in in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

- d. Click **Add**. The Consume Adapter Service Add-in opens.
2. From the **Select a binding** drop-down list, select **oracleEBSBinding** and click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

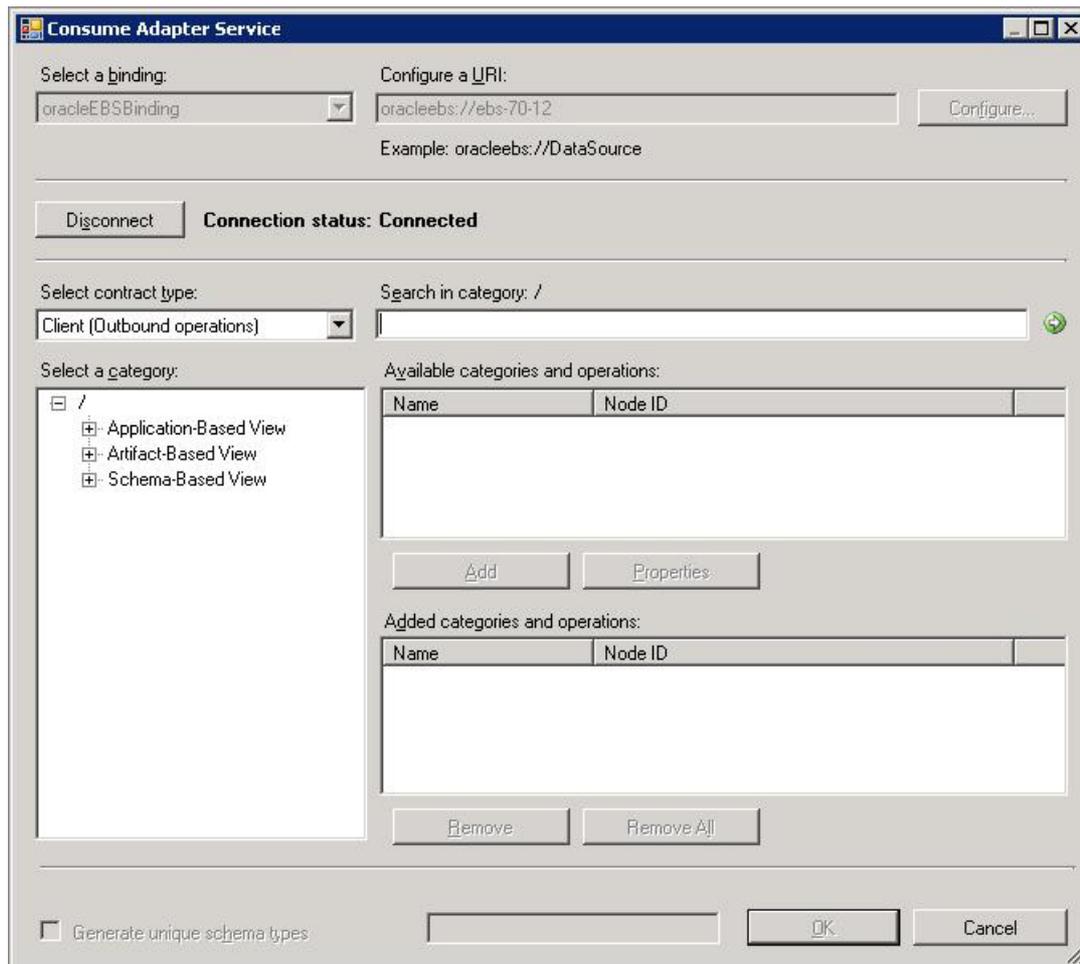
4. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).

 **Note**

If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
6. Click **OK**.
7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on the Oracle E-Business Suite and the Oracle database. For more information on how the metadata is categorized under the various nodes, see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#)

See Also

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Other Resources

[Connecting to the Oracle E-Business Suite in Visual Studio](#)

Connecting to Oracle E-Business Suite in Visual Studio Using Add Adapter Metadata Wizard

The Oracle E-Business adapter is also exposed as a BizTalk adapter and, therefore, you can use the Add Adapter Metadata Wizard to generate schema for the operations you want to perform on Oracle E-Business Suite using the adapter.

Connecting to Oracle E-Business Suite Using the Add Adapter Metadata Wizard

Perform the following steps to connect to Oracle E-Business Suite using the Add Adapter Metadata Wizard.

To connect to Oracle E-Business Suite

1. To connect using the Add Adapter Metadata Wizard in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

- d. Click **Add**. The Add Adapter Metadata Wizard opens.
- e. In the Add Adapter Metadata Wizard, select **WCF-OracleEBS**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-OracleEBS port configured in BizTalk, select the port from the Port list.

- f. Click **Next**.
2. From the **Select a binding** drop-down list, select **oracleEBSBinding** and click **Configure**.
 3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.

<p>To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"</p>	<p>Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.</p>
---	--

- Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).

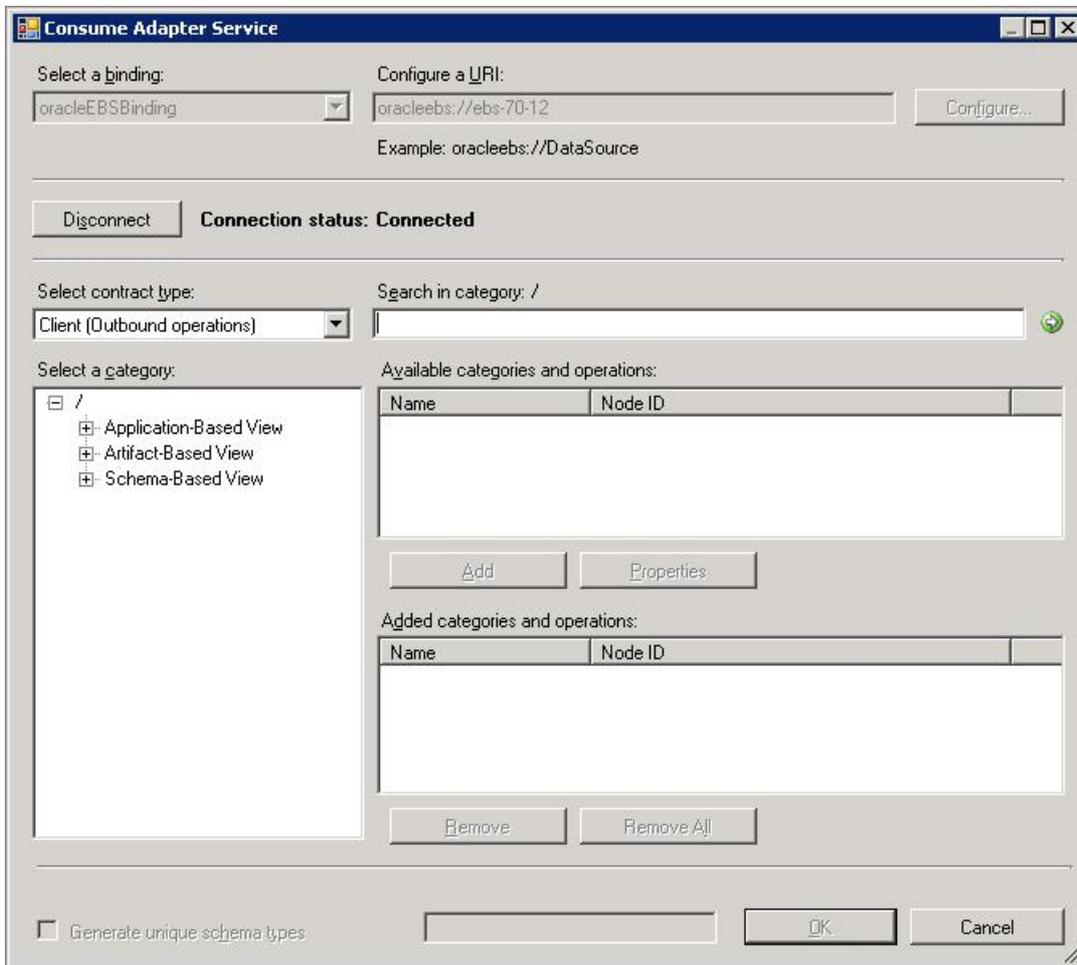
Note
 If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

- Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Note
 If you are generating metadata using Add Adapter Metadata Wizard and you selected an existing WCF-OracleEBS send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

- Click **OK**.
- Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on the Oracle E-Business Suite and the Oracle database. For more information on how the metadata is categorized under the various nodes, see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#)

See Also

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Other Resources

[Connecting to the Oracle E-Business Suite in Visual Studio](#)

Connecting to Oracle E-Business Suite in Visual Studio Using Add Adapter Service Reference Plug-in

To connect to Oracle E-Business Suite using the Oracle E-Business adapter in a .NET programming solution, you must use the Add Adapter Service Reference Visual Studio Plug-in. This topic provides instructions on how to use the Add Adapter Service Reference Plug-in.

Connecting to Oracle E-Business Suite Using the Add Adapter Service Reference Plug-in

Perform the following steps to connect to Oracle E-Business Suite using the Add Adapter Service Reference Plug-in.

To connect to Oracle E-Business Suite

1. To connect using the Add Adapter Service Reference Plug-in in a programming solution:
 - a. Create a project using Visual Studio.
 - b. Right-click the project in Solution Explorer, and then click **Add Adapter Service Reference**. The Add Adapter Service Reference Plug-in opens.
2. From the **Select a binding** drop-down list, select **oracleEBSBinding** and click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

4. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).

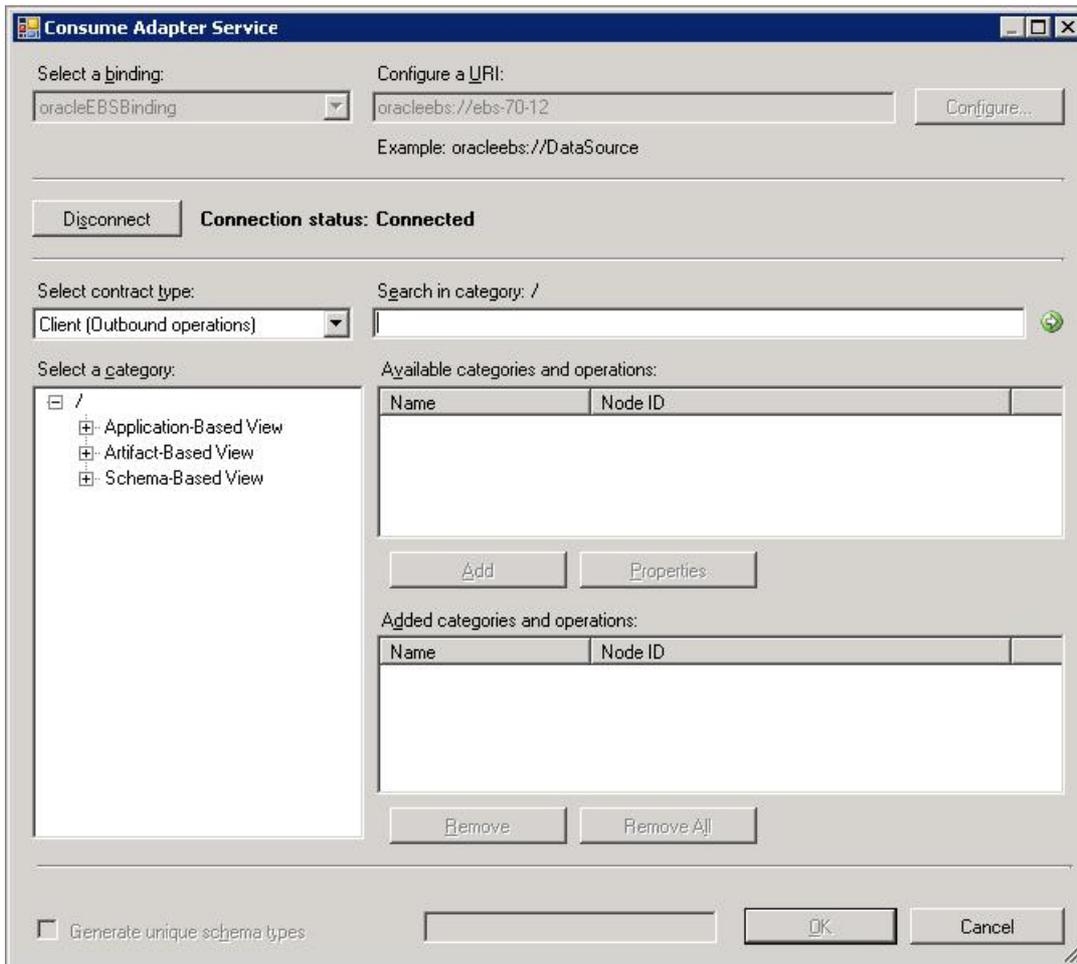
Note

If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

6. Click **OK**.
7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established. The graphical user interface is the same for the Add Adapter Service Reference Plug-in.



The Add Adapter Service Reference Plug-in displays different nodes containing various operations that can be performed on the Oracle E-Business Suite and the Oracle database. For more information on how the metadata is categorized under the various nodes, see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#).

See Also

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Other Resources

[Connecting to the Oracle E-Business Suite in Visual Studio](#)

Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations

This section provides information about how to use the Consume Adapter Service BizTalk Project Add-in and the Add Adapter Service Reference Visual Studio Plug-in. By using these Visual Studio components, you can:

- Browse for operations to retrieve metadata.
- Search for operations to retrieve metadata.
- Add message schemas for selected operations and port binding configuration files to a BizTalk Server project when using the Consume Adapter Service Add-in.
- Add a WCF client class or a WCF service contract (interface) for selected operations and a configuration file (app.config) to a non-BizTalk programming project when using the Add Adapter Service Reference Plug-in.

How is the Metadata Categorized?

The Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in gives three different views of the artifacts available in the Oracle E-Business Suite server you connect to—**Application-based view**, **Artifact-based view**, and the **Schema-based view**. Why do you need three different views for the same set of artifacts? The following table lists the reasons why you should use a specific view.

View	When do you use it
Application-based view	This view is categorized by the Oracle E-Business Suite application names. Use this view when you know which application contains the artifacts you want to work with.
Artifact-based view	<p>This view is categorized by the Oracle E-Business Suite artifacts. Use this view when you know which Oracle E-Business Suite artifact you want to work with but you are not sure which application the artifact belongs to. Using this view, you can search for a specific artifact across all Oracle E-Business Suite applications.</p> <p>The artifact-based view also lists the artifacts in the underlying Oracle database such as PL-SQL APIs, tables, views, functions, and procedures. These artifacts are further categorized based on the schema they belong to. So, another use of the artifact-based view is to use artifacts that belong to your schema as well as the artifacts that belong to other schemas. This view also enables you to search for artifacts across all schemas.</p>
Schema-based view	This view is categorized by the schemas available in the underlying Oracle database. Use this view when you know which schema has the artifacts you want to work with. Within this view, the artifacts are categorized as PL-SQL APIs, procedures, functions, tables, and views.

For more information on how the Oracle E-Business Suite artifacts are categorized, see

[How Does the Adapter Surface Oracle E-Business Suite Metadata?](#) Another key reason for organizing the artifacts in different views is the ease to search for specific artifacts. For more information on how you can search for artifacts, see [Searching for Operations](#).

◆ Important

The nodes show up based on the connection URI you specify while establishing a connection. If you specify credentials that do not have permissions on the Oracle E-Business Suite artifacts, you cannot use the artifacts in the **Application-based view**. Also, the **Artifact-based view** does not list the artifacts belonging to Oracle E-Business Suite.

In This Section

- [Browsing for Operations Under the Application-Based View](#)

- [Browsing for Operations Under the Artifact-Based View](#)
- [Browsing for Operations Under the Schema-Based View](#)
- [Searching for Operations](#)
- [Retrieving Metadata for Operations](#)

See Also

Other Resources

[Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#)

Browsing for Operations Under the Application-Based View

You can use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in to browse for outbound and inbound operations that can be performed on Oracle E-Business Suite using the Oracle E-Business adapter. This topic provides information on how to browse for outbound and inbound operations under the application-based view.

Note

The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so both components are covered in the same topics.

Prerequisites

You must connect to the Oracle E-Business Suite before you can browse metadata for target operations. For information about how to connect to the Oracle database when you use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

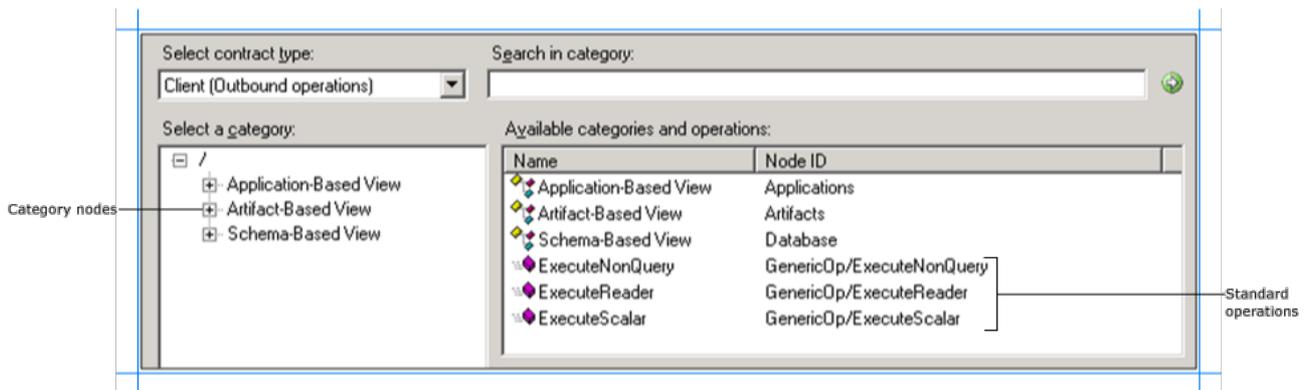
Browsing for Outbound Operations

Perform the following steps to browse the outbound operations under the application-based view.

To browse metadata for outbound operations under the Application-Based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, select **Client (Outbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



Note

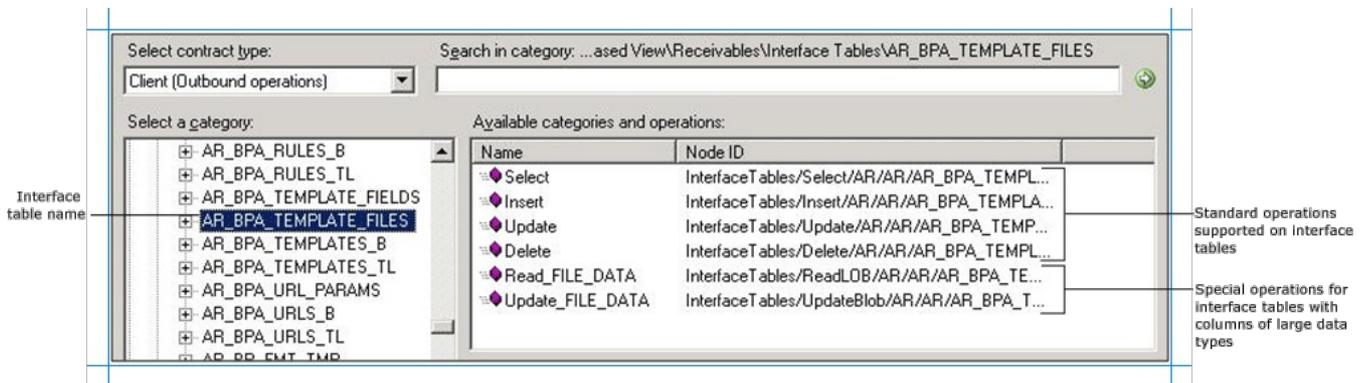
The standard operations such as ExecuteReader, ExecuteScalar, and ExecuteNonQuery are available at the root level. For more information about these operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#). For instructions on how to execute these operations using the Oracle E-Business adapter, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server](#).

4. Expand the **Application-Based View** node to see all the Oracle E-Business suite applications available on the server you connected to. Expand an application to see categories for interface tables, interface views, concurrent programs, and request sets available for that application.

Tip

You can directly go to the "immediate" category node or subcategory nodes in the tree, by typing the name of the artifact in while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Alert** node, keep the focus on the **Application-Based View** node, and the n type **Alert**.

5. Expand the **Interface Tables** node to see the interface tables for the Oracle application. Click an interface table to see the list of operations available for the table in the **Available categories and operations** box.

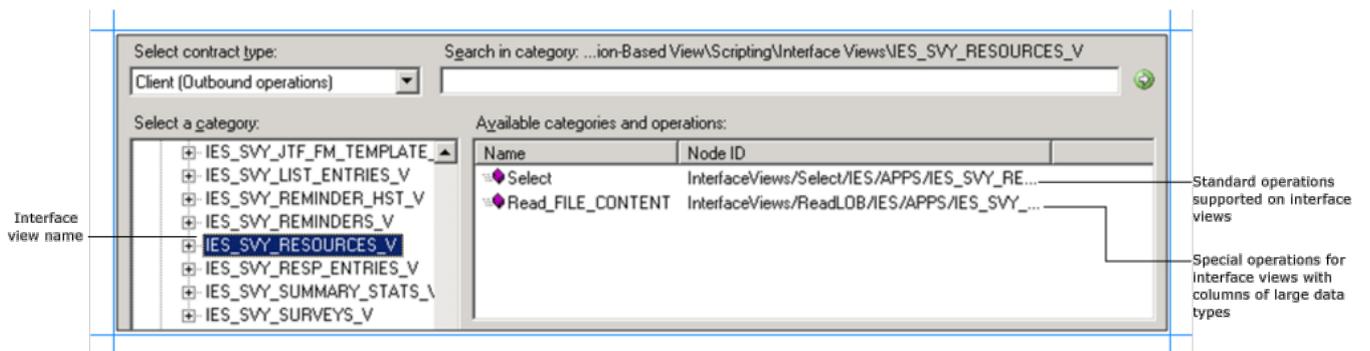


Note

If an interface table contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the interface table has a column, FILE_DATA, of type BLOB, the adapter exposes a **Read_FILE_DATA** operation. If an interface table has more than one column of type BLOB, CLOB, NCLOB, and BFILE the adapter will expose as many number of Read_<LOBColumnName> operations.

Similarly, if an interface table contains columns of type BLOB, CLOB, or NCLOB the adapter also exposes a specific operation to update data into such columns. The name of such operations are Update_<LOBColumnName>. For example, if the interface table has a column, FILE_DATA, of type BLOB, the adapter exposes an **Update_FILE_DATA** operation. If an interface table has more than one column of type BLOB, CLOB, and NCLOB the adapter will expose as many number of Update_<LOBColumnName> operations. Note that the update operation is not supported on columns of type BFILE.

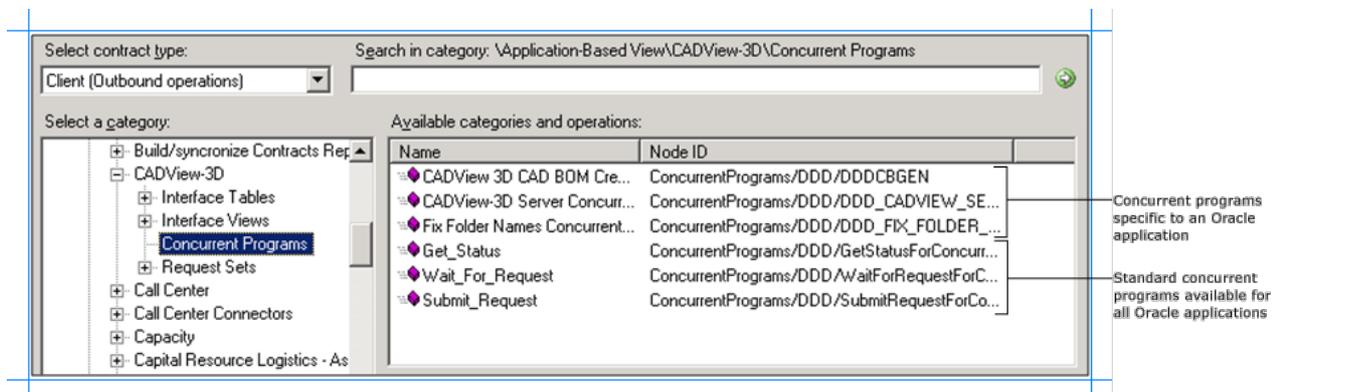
- Expand the **Interface Views** node to see the interface views for the Oracle application. Click an interface view to see the list of operations available for the view in the **Available categories and operations** box.



Note

If an interface view contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the interface view has a column, FILE_CONTENT, of type BLOB, the adapter exposes a **Read_FILE_CONTENT** operation. If an interface view has more than one column of type BLOB, CLOB, NCLOB, or BFILE the adapter will expose as many number of Read_<LOBColumnName> operations. Note that Update_<LOBColumnName> operations are not supported on views.

- Click the **Concurrent Programs** node to see the concurrent programs for an application in the **Available categories and operations** box.

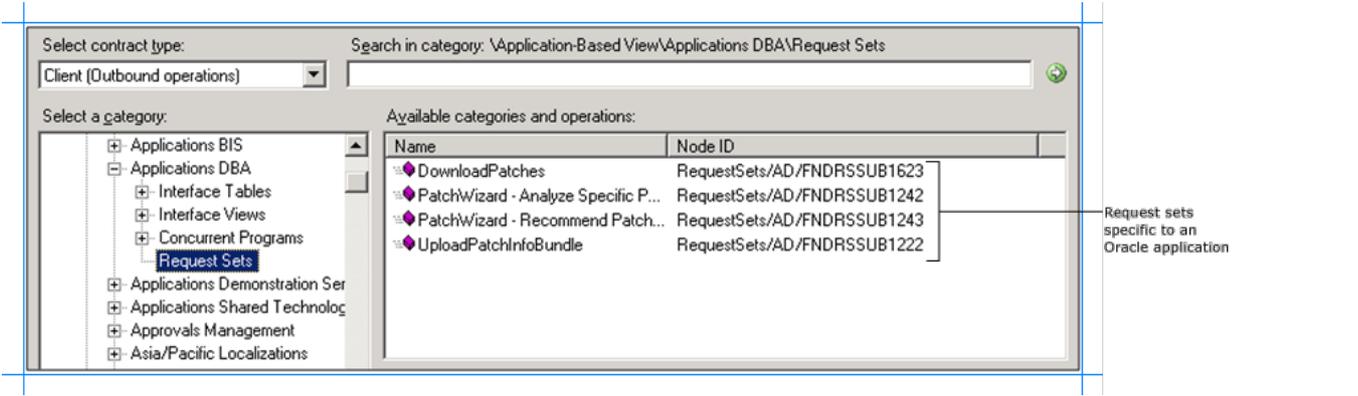


This figure shows the concurrent programs specific to an Oracle application and the standard concurrent programs for all Oracle applications.

Important

The Consume Adapter Service Add-in (or the Add Adapter Service Reference Plug-in) displays friendly names of concurrent programs. However, the metadata for the concurrent program has the actual name of the concurrent program. For example, the Receivables application contains a “Customer Interface” concurrent program. However, the metadata has the concurrent program name as RACUST, which is the actual name of the concurrent program.

8. Click the **Request Sets** node to see the request sets for an application in the **Available categories and operations** box.



Important
 The Consume Adapter Service Add-in (or the Add Adapter Service Reference Plug-in) displays friendly names of request sets. However, the metadata for the request set has the actual name of the request set. For example, the Applications DBA application contains a “DownloadPatches” request set. However, the metadata has the request set name as FNDRSSUB1623, which is the actual name of the request set.

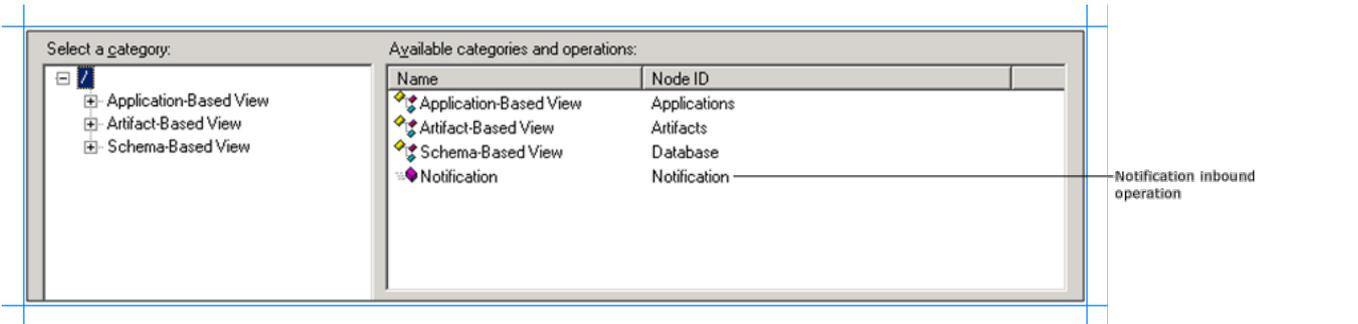
Browsing for Inbound Operations

Perform the following steps to browse the inbound operations under the application-based view.

To browse metadata for inbound operations under the Application-based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for inbound operations select **Service (Inbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



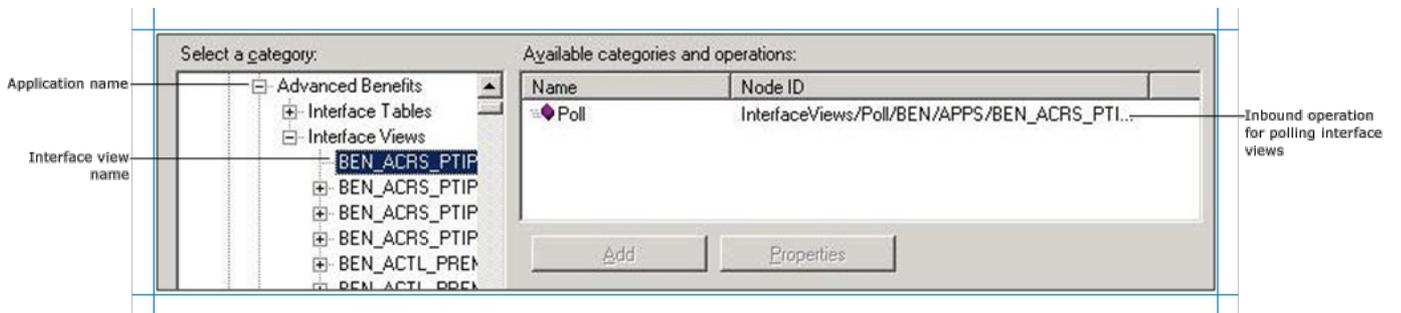
The inbound operation, **Notification**, is also available at the root level.

4. Expand the **Application-Based View** node to see all the Oracle E-Business suite applications available on the server you connected to. Expand an application to see categories for interface tables and interface views.

Tip
 You can directly go to the “immediate” category node or subcategory nodes in the tree, by typing the name of the artifact in while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Alert** node, keep the focus on the **Application-Based View** node, and the n type **Alert**.

5. Expand an Oracle application to see categories for interface tables and interface views available for that application. Expand the **Interface Tables** and **Interface Views** nodes to see the interface tables and interface views for the Oracle application. Click an interface table or interface view to see the inbound operation available for the table or view in the **Available categories and operations** box.

In the following figure, an interface view is selected in the **Select a category** box and the inbound operation supported on the view is listed in the **Available categories and operations** box.



Note
The Oracle E-Business adapter does not surface concurrent programs and request sets for inbound operations.

See Also

Other Resources

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Browsing for Operations Under the Artifact-Based View

You can use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in to browse for outbound and inbound operations that can be performed on Oracle E-Business Suite using the Oracle E-Business adapter. This topic provides information on how to browse for outbound and inbound operations under the artifact-based view.

Note
The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so both components are covered in the same topics.

Prerequisites

You must connect to the Oracle E-Business Suite before you can browse metadata for target operations. For information about how to connect to the Oracle database when you use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

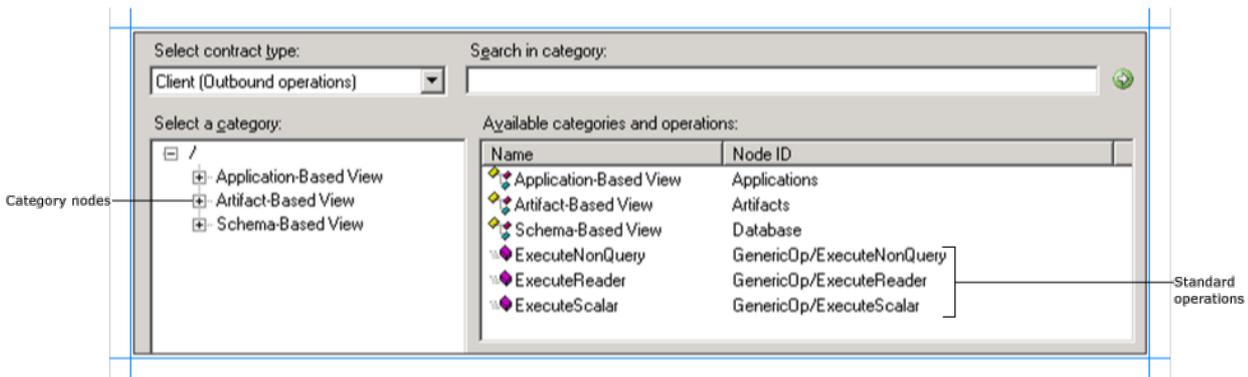
Browsing for Outbound Operations

Perform the following steps to browse the outbound operations under the artifact-based view.

To browse metadata for outbound operations under the Artifact-Based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for outbound operations select **Client (Outbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



Note
The standard operations such as ExecuteReader, ExecuteScalar, and ExecuteNonQuery are available at the root level. For more information about these operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#). For instructions on how to execute these operations using the Oracle E-Business adapter, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server](#).

4. Expand the **Artifact-Based View** node to see category for artifacts, both for the Oracle E-Business Suite and the underlying database. Each category is further categorized based on the application it belongs to (for Oracle-E-Business Suite artifacts such as interface table, interface views, concurrent programs, and request sets) or the schema it belongs to (for Oracle database artifacts such as PL-SQL APIs, procedures, functions, tables, and views).

Tip
You can directly go to the “immediate” category node or subcategory nodes in the tree, by typing the name of the artifact while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Procedures** node, keep the focus on the **Artifact-Based View** node, and then type **Procedures**.

5. Expand the **Interface Tables** node to see all the Oracle E-Business Suite applications. Expand an Oracle E-Business suite application to list all the interface tables belonging to that application. Click an interface table name to see operations available for the table in the **Available categories and operations** box.

Note
If an interface table contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the interface table has a column, FILE_DATA, of type BLOB, the adapter exposes a **Read_FILE_DATA** operation. If an interface table has more than one column of type BLOB, CLOB, NCLOB, and BFILE the adapter will expose as many number of Read_<LOBColumnName> operations.

Similarly, if an interface table contains columns of type BLOB, CLOB, or NCLOB the adapter also exposes a specific operation to update data into such columns. The name of such operations are Update_<LOBColumnName>. For example, if the interface table has a column, FILE_DATA, of type BLOB, the adapter exposes an **Update_FILE_DATA** operation. If an interface table has more than one column of type BLOB, CLOB, and NCLOB the adapter will expose as many number of Update_<LOBColumnName> operations. Note that the update operation is not supported on columns of type BFILE.

- Expand the **Interface Views** node to see all the Oracle E-Business Suite applications. Expand an Oracle E-Business suite application to list all the interface views belonging to that application. Click an interface view name to see operations available for the view in the **Available categories and operations** box.

Note

If an interface view contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The names of such operations are Read_<LOBColName>. For example, if the interface view has a column, FILE_CONTENT, of type BLOB, the adapter exposes a **Read_FILE_CONTENT** operation. If an interface view has more than one column of type BLOB, CLOB, NCLOB, or BFILE the adapter will expose a s many number of Read_<LOBColName> operations. Note that Update_<LOBColName> operations are not supported on views.

- Expand the **Concurrent Programs** node to see all the Oracle E-Business Suite applications. Click an Oracle E-Business suite application to list all the concurrent programs belonging to that application in the **Available categories and operations** box.

Important

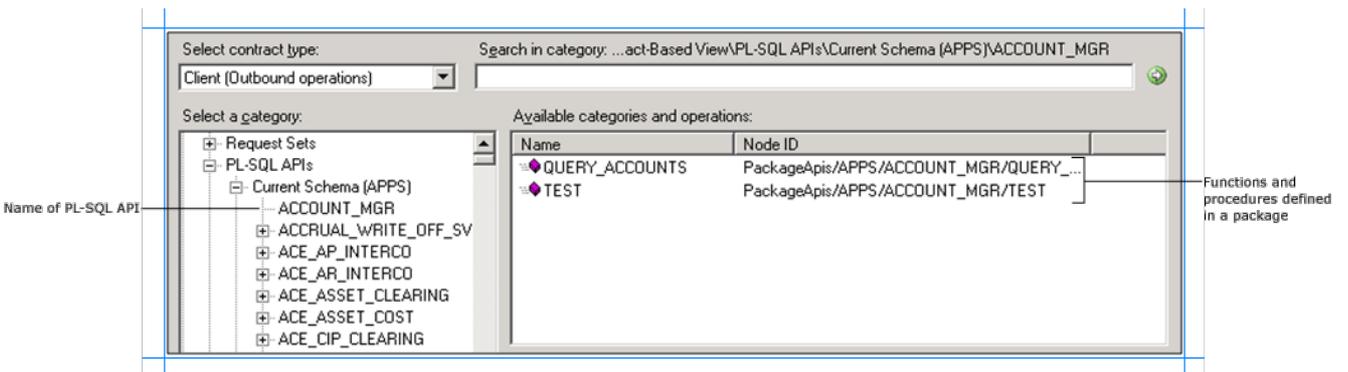
The Consume Adapter Service Add-in (or the Add Adapter Service Reference Plug-in) displays friendly names of concurrent programs. However, the metadata for the concurrent program has the actual name of the concurrent program. For example, the Receivables application contains a "Customer Interface" concurrent program. However, the metadata has the concurrent program name as RACUST, which is the actual name of the concurrent program.

- Expand the **Request Sets** node to see all the Oracle E-Business Suite applications. Click an Oracle E-Business suite application to list all the request sets belonging to that application in the **Available categories and operations** box.

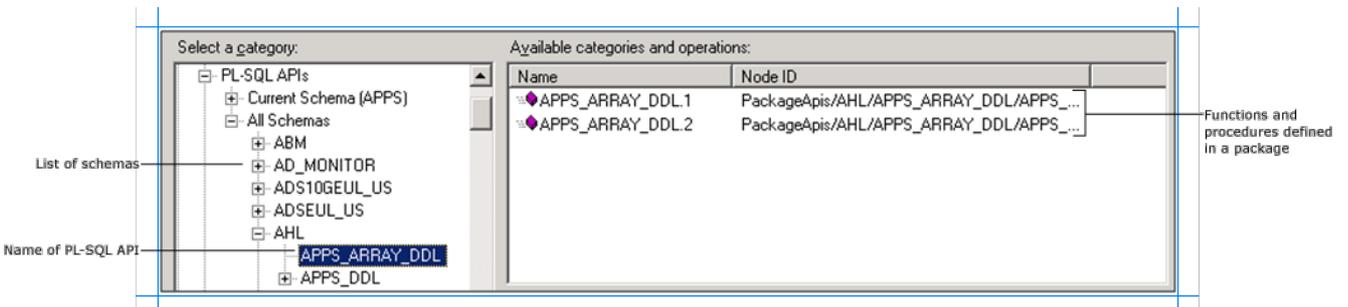
Important

The Consume Adapter Service Add-in (or the Add Adapter Service Reference Plug-in) displays friendly names of request sets. However, the metadata for the request set has the actual name of the request set. For example, the Applications DBA application contains a "DownloadPatches" request set. However, the metadata has the request set name as FNDRSSUB1623, which is the actual name of the request set.

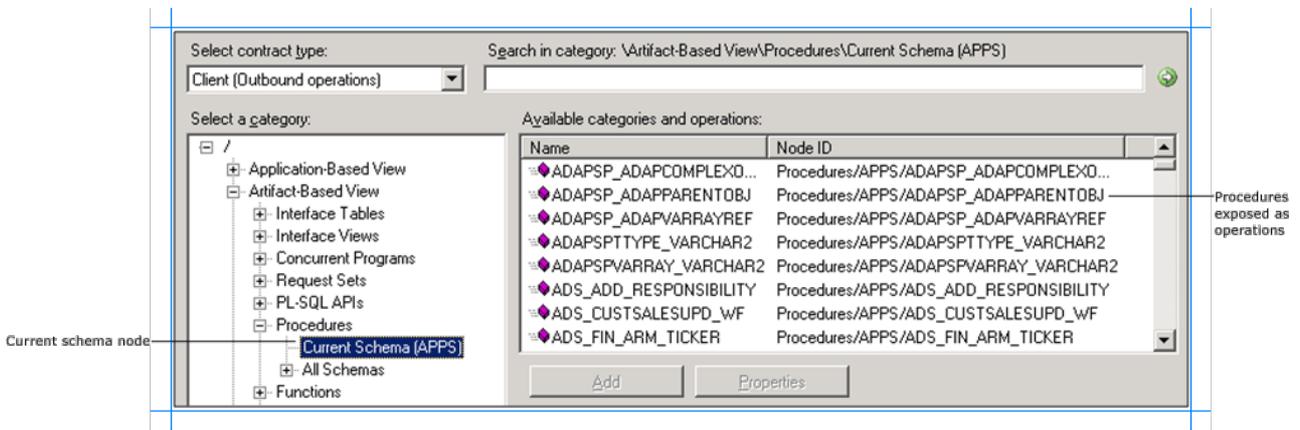
- Expand the **PL-SQL APIs** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all packages defined for that schema. Click a package name to see the functions and procedures within the package in the **Available categories and operations** box.



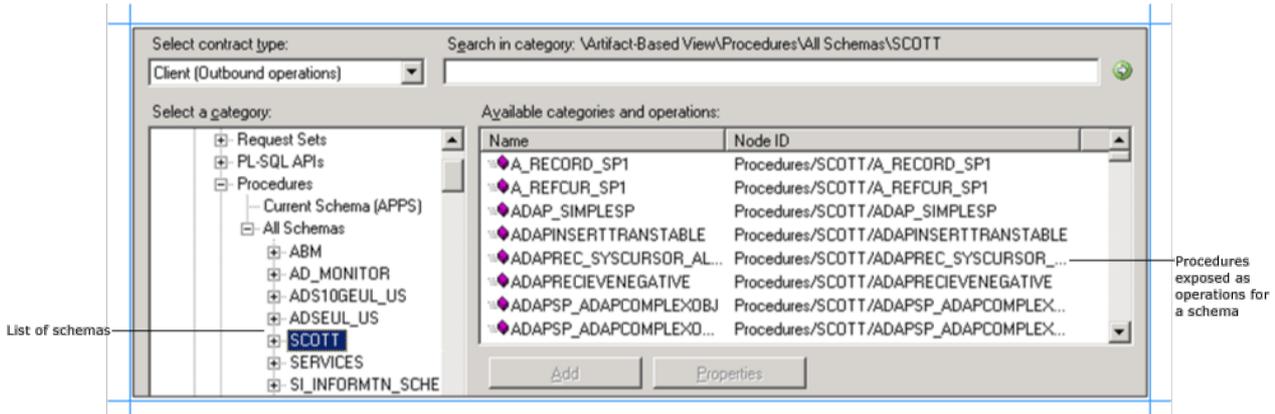
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of packages defined for that schema. Click a package name to see the functions and procedures within the package in the **Available categories and operations** box.



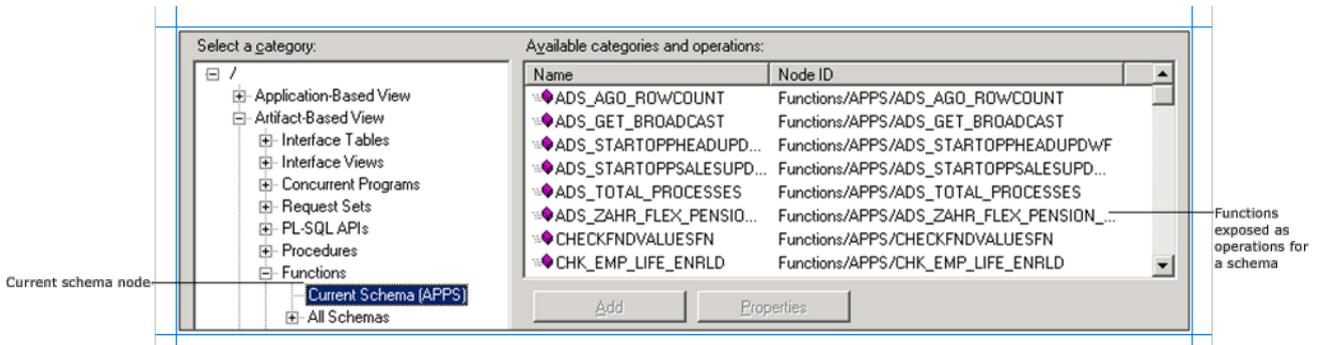
- Expand the **Procedures** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Click the **Current Schema (<schema name>)** node to see all the procedures defined for that schema in the **Available categories and operations** box.



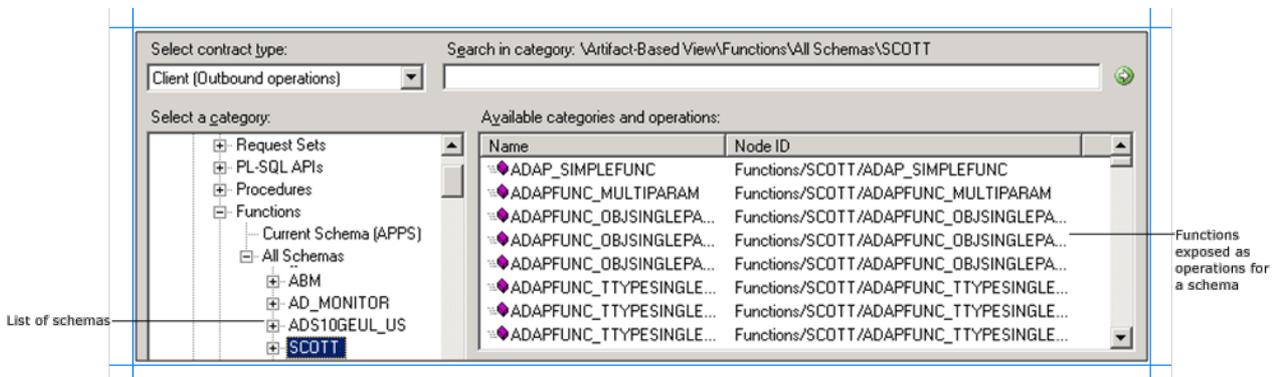
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Click a schema node to see a list of procedures defined for that schema in the **Available categories and operations** box.



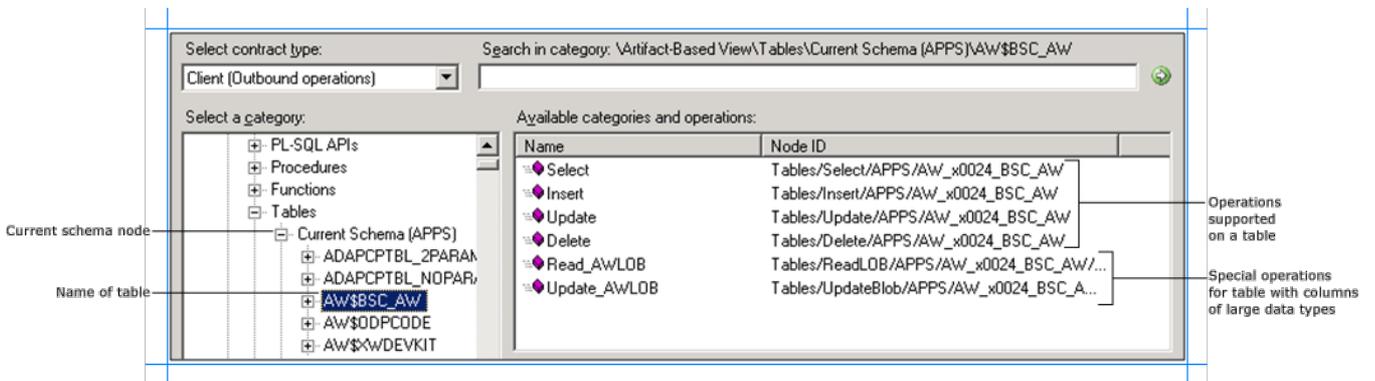
- Expand the **Functions** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Click the **Current Schema (<schema name>)** node to see all the functions defined for that schema in the **Available categories and operations** box.



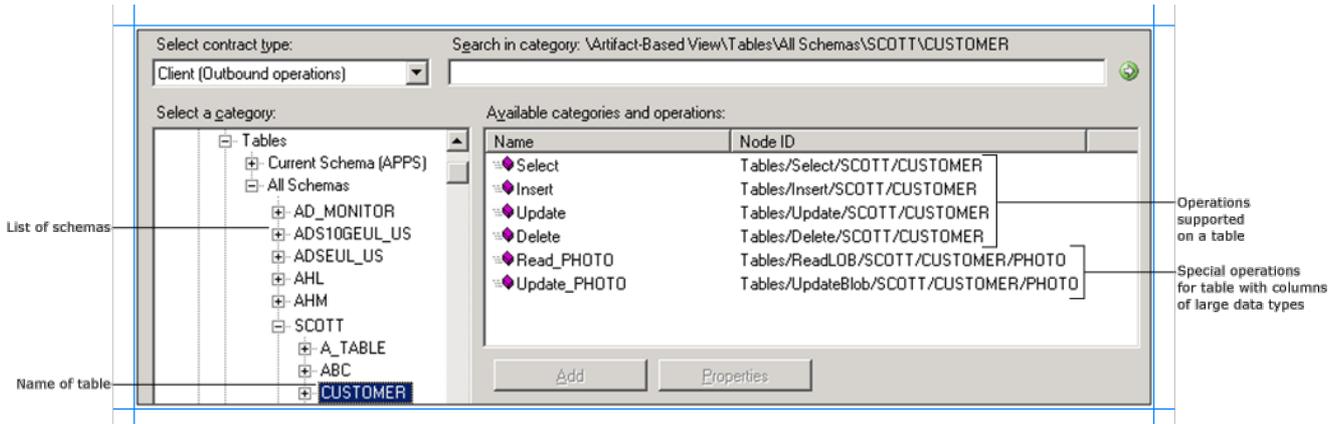
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Click a schema node to see a list of functions defined for that schema in the **Available categories and operations** box.



- Expand the **Tables** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all the tables defined for that schema. Click a table name to see the operations supported on that table in the **Available categories and operations** box.



Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of tables defined for that schema. Click a table name to see the operations supported on that table in the **Available categories and operations** box.

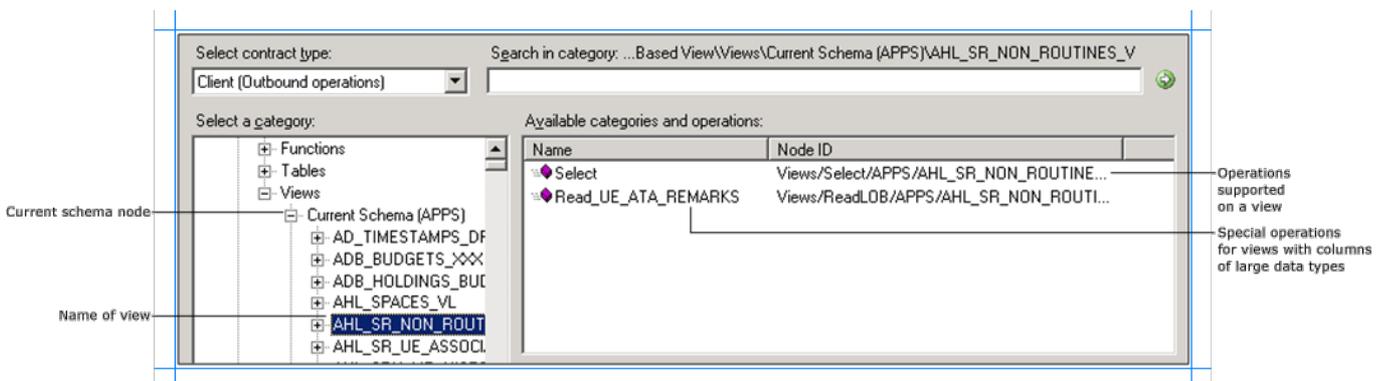


Note

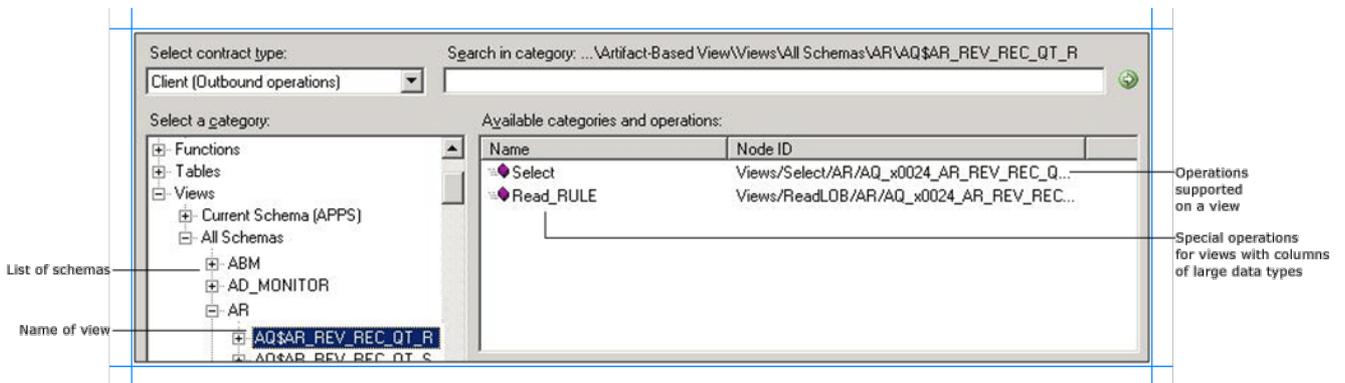
If a table contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the table has a column, PHOTO, of type BLOB, the adapter exposes a **Read_PHOTO** operation. If a table has more than one column of type BLOB, CLOB, NCLOB, and BFILE the adapter will expose as many number of Read_<LOBColumnName> operations.

Similarly, if a table contains columns of type BLOB, CLOB, or NCLOB the adapter also exposes a specific operation to update data into such columns. The name of such operations are Update_<LOBColumnName>. For example, if the table has a column, PHOTO, of type BLOB, the adapter exposes an **Update_PHOTO** operation. If a table has more than one column of type BLOB, CLOB, and NCLOB the adapter will expose as many number of Update_<LOBColumnName> operations. Note that the update operation is not supported on columns of type BFILE.

- Expand the **Views** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all the views defined for that. Click a view name to see the operations supported on that view in the **Available categories and operations** box.



Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of views defined for that schema. Click a view name to see the operations supported on that view in the **Available categories and operations** box.



Note
 If a view contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the view has a column, RULE, of type BLOB, the adapter exposes a **Read_RULE** operation. If a view has more than one column of type BLOB, CLOB, NCLOB, or BFILE the adapter will expose as many number of Read_<LOBColumnName> operations. Note that Update_<LOBColumnName> operations are not supported on views.

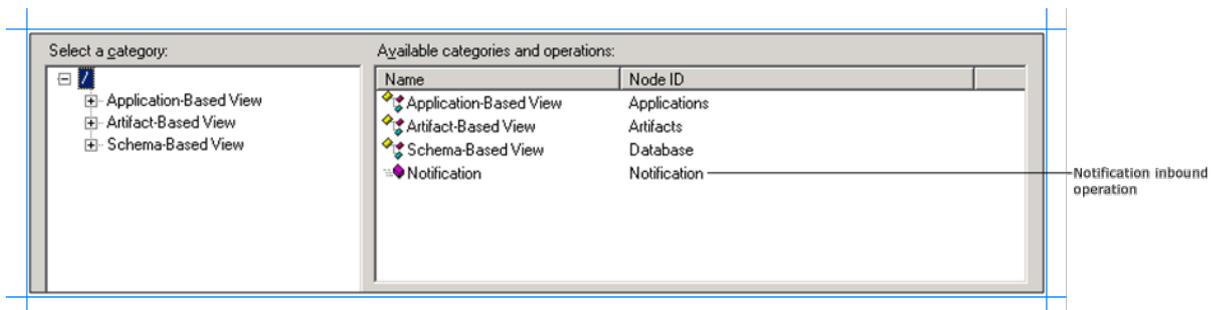
Browsing for Inbound Operations

Perform the following steps to browse the inbound operations under the artifact-based view.

To browse metadata for inbound operations under the Artifact-based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for inbound operations select **Service (Inbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



The inbound operation, **Notification**, is also available at the root level.

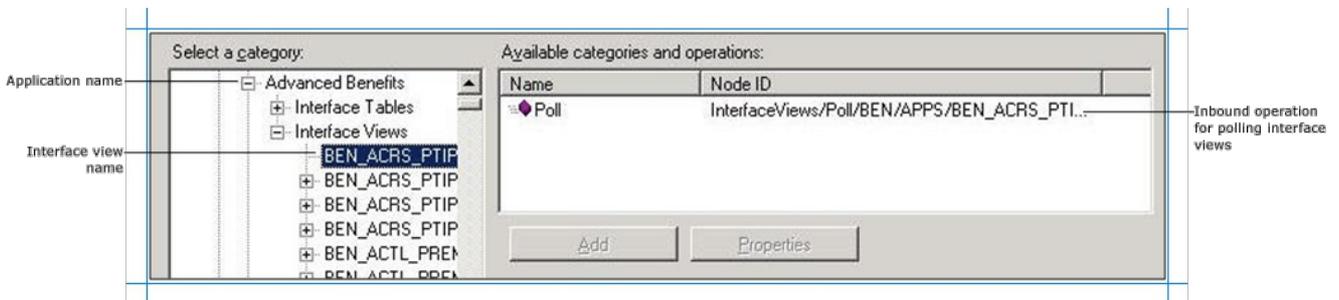
4. Expand the **Artifact-Based View** node to see category for artifacts, both for the Oracle E-Business Suite and the underlying database. Each category is further categorized based on the application it belongs to (for Oracle E-Business Suite artifacts such as interface table and interface views) or the schema it belongs to (for Oracle database artifacts such as PL-SQL APIs, procedures, functions, tables, and views).

Tip
 You can directly go to the “immediate” category node or subcategory nodes in the tree, by typing the name of the artifact while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Procedures** node, keep the focus on the **Artifact-Based View** node, and then type **Procedures**.

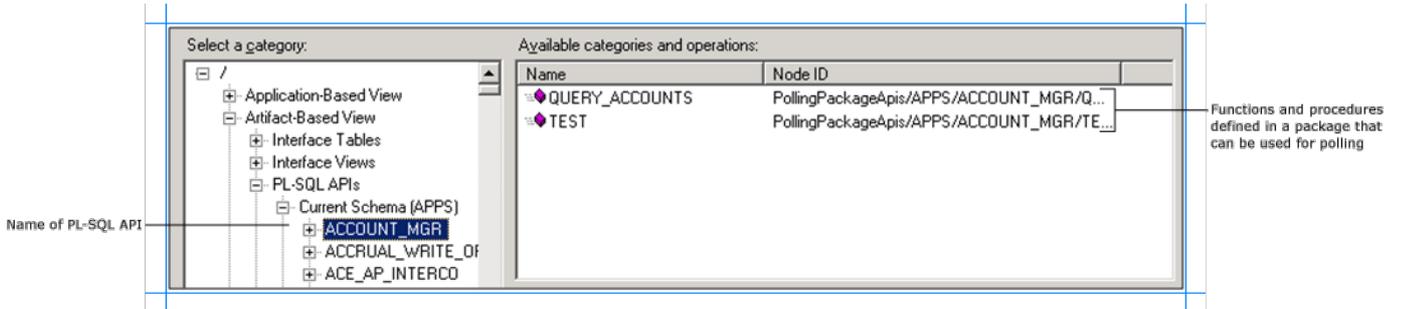
Note
 The Oracle E-Business adapter does not surface concurrent programs and request sets for inbound operations.

5. Expand an Oracle application to see categories for interface tables and interface views available for that application. Expand the **Interface Tables** and **Interface Views** nodes to see the interface tables and interface views for the Oracle application. Click an interface table or interface view to see the inbound operation available for the table or view in the **Available categories and operations** box.

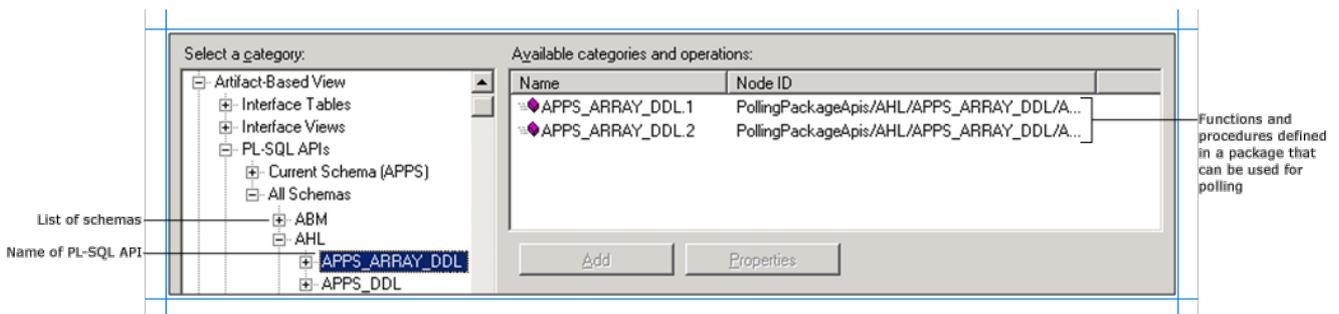
In the following figure, an interface view is selected in the **Select a category** box and the inbound operation supported on the view is listed in the **Available categories and operations** box.



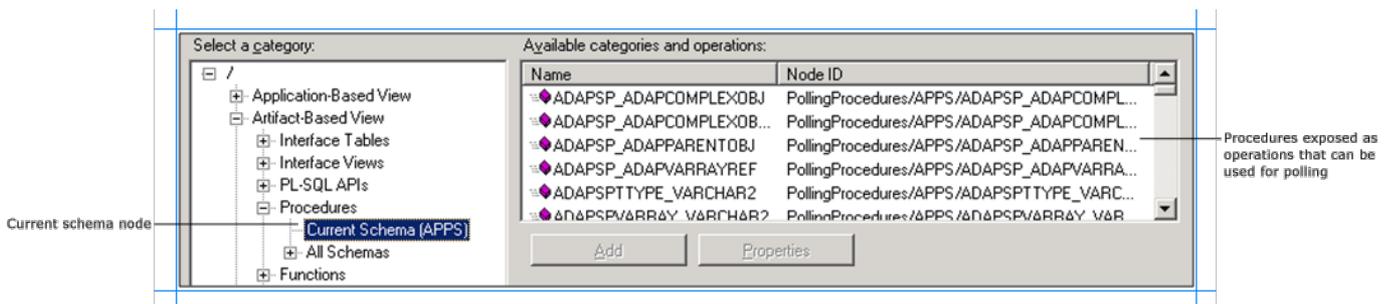
- Expand the **PL-SQL APIs** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all packages defined for that schema. Click a package name to see the functions and procedures within the package in the **Available categories and operations** box. Each of the listed functions and procedures can be used to poll the Oracle database.



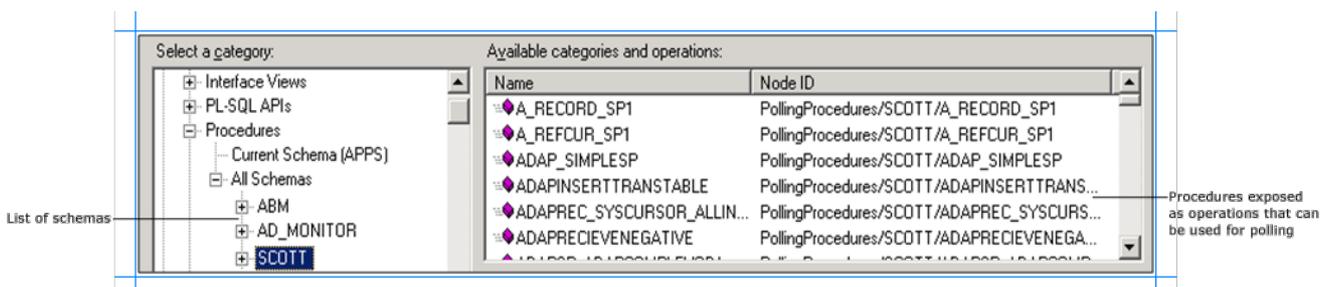
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of packages defined for that schema. Click a package name to see the functions and procedures within the package in the **Available categories and operations** box. Each of the listed functions and procedures can be used to poll the Oracle database.



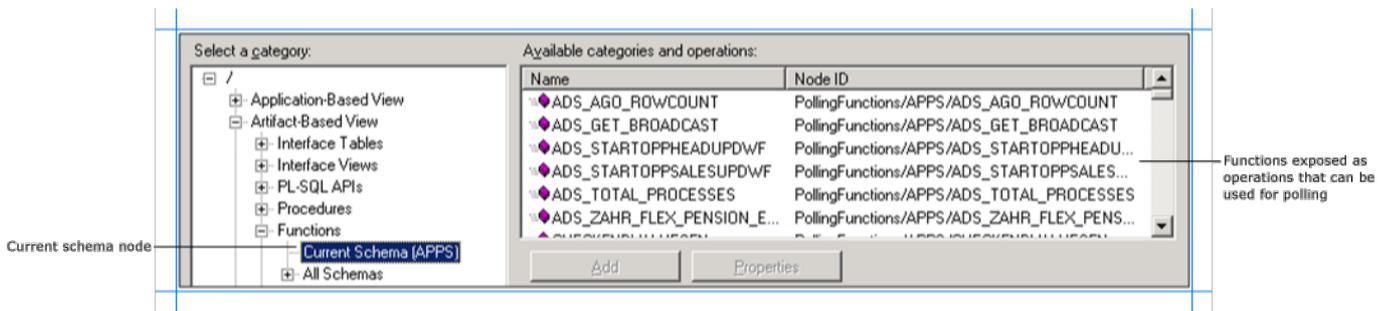
- Expand the **Procedures** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Click the **Current Schema (<schema name>)** node to see all the procedures defined for that schema in the **Available categories and operations** box. Each of the listed procedures can be used to poll the Oracle database.



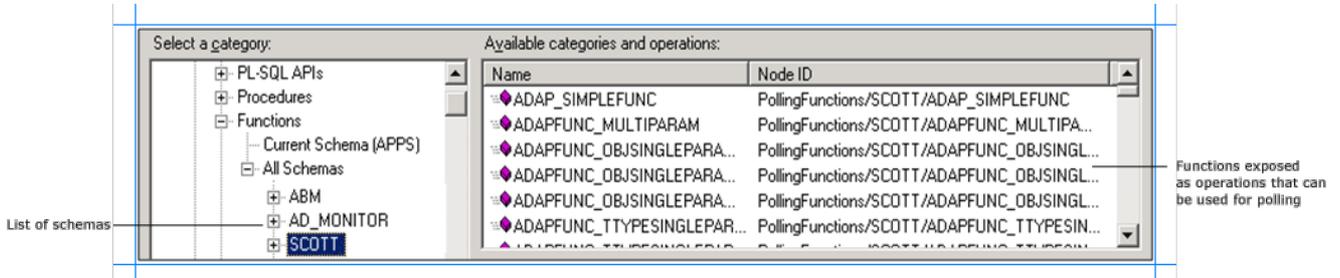
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Click a schema node to see a list of procedures defined for that schema in the **Available categories and operations** box. Each of the listed procedures can be used to poll the Oracle database.



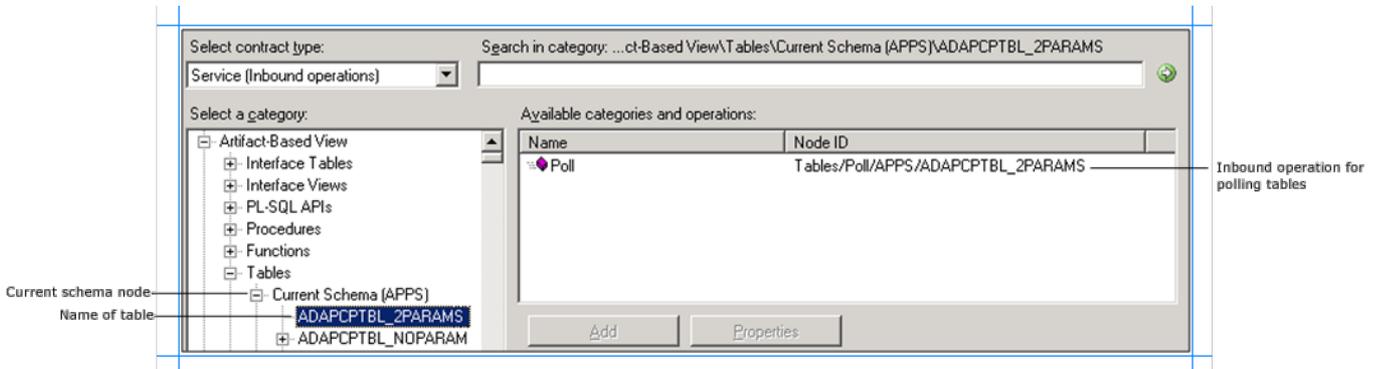
- Expand the **Functions** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Click the **Current Schema (<schema name>)** node to see all the functions defined for that schema in the **Available categories and operations** box. Each of the listed functions can be used to poll the Oracle database.



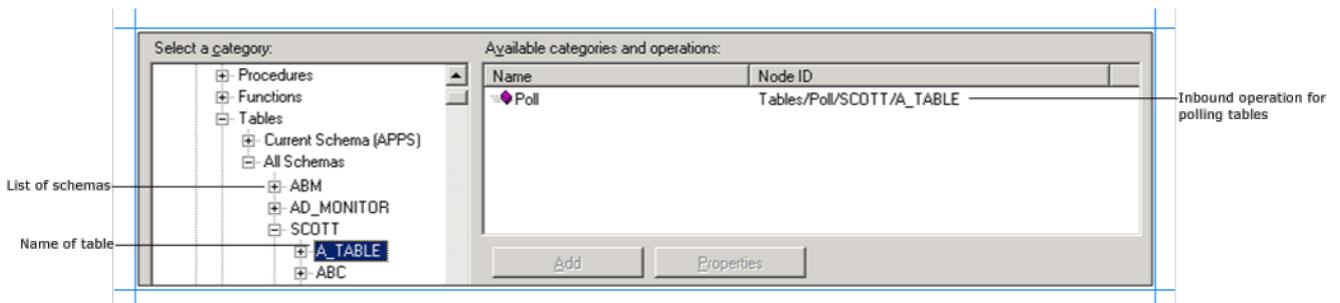
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Click a schema node to see a list of functions defined for that schema in the **Available categories and operations** box. Each of the listed functions can be used to poll the Oracle database.



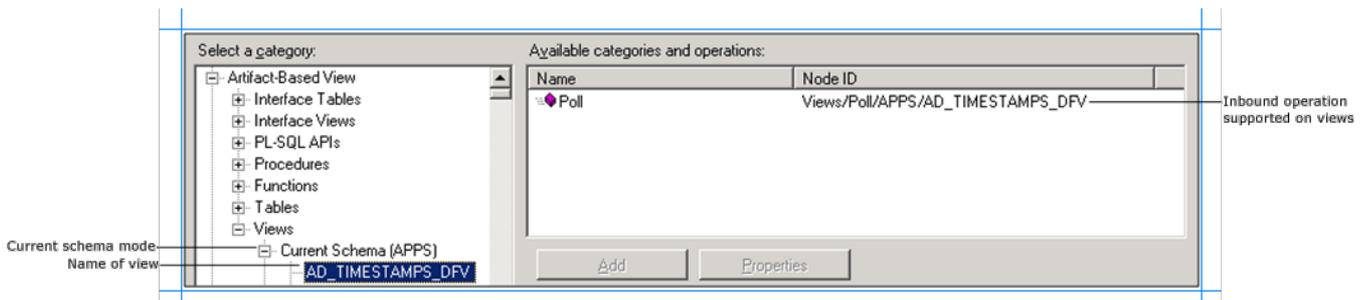
- Expand the **Tables** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all the tables defined for that schema. Click a table name to see the **Poll** inbound operation supported on that table in the **Available categories and operations** box.



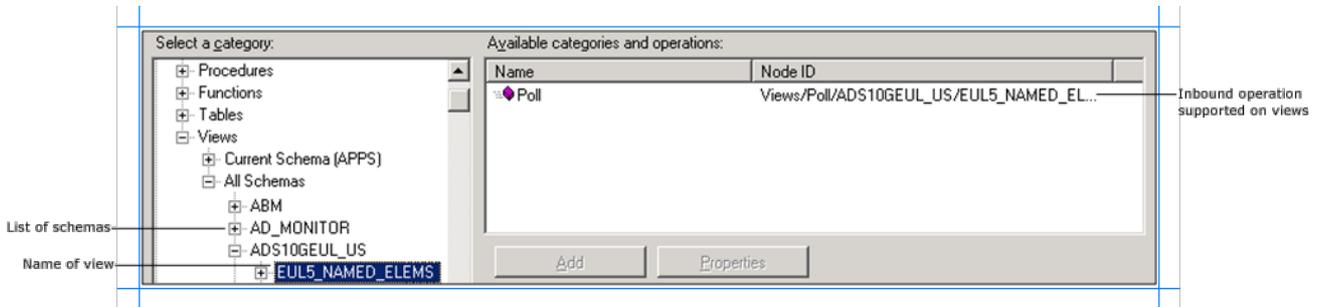
Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of tables defined for that schema. Click a table name to see the **Poll** inbound operation supported on that table in the **Available categories and operations** box.



- Expand the **Views** node to see category nodes for the current user schema (with which you login) and all other schemas defined in the underlying Oracle database. Expand the **Current Schema (<schema name>)** node to see all the views defined for that schema. Click a view name to see the **Poll** inbound operation supported on that view in the **Available categories and operations** box.



Similarly, expand the **All Schemas** node to see a list of all the schemas defined in the Oracle database. Expand a schema node to see a list of views defined for that schema. Click a view name to see the **Poll** inbound operation supported on that view in the **Available categories and operations** box.



See Also

Other Resources

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Browsing for Operations Under the Schema-Based View

You can use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in to browse for outbound and inbound operations that can be performed on Oracle E-Business Suite using the Oracle E-Business adapter. This topic provides information on how to browse for outbound and inbound operations under the schema-based view.

Note

The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so both components are covered in the same topics.

Prerequisites

You must connect to the Oracle E-Business Suite before you can browse metadata for target operations. For information about how to connect to the Oracle database when you use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

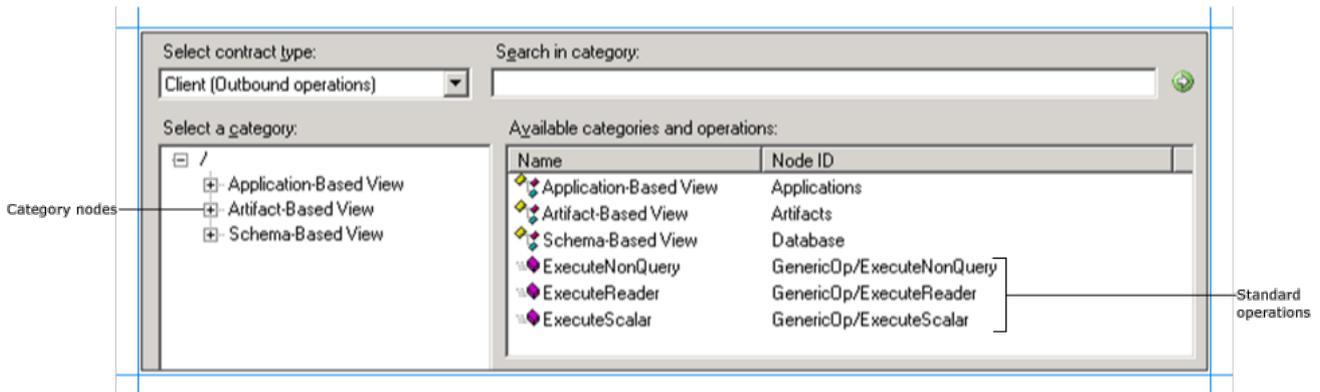
Browsing for Outbound Operations

Perform the following steps to browse the outbound operations under the schema-based view.

To browse metadata for outbound operations under the Schema-Based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for outbound operations select **Client (Outbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



Note

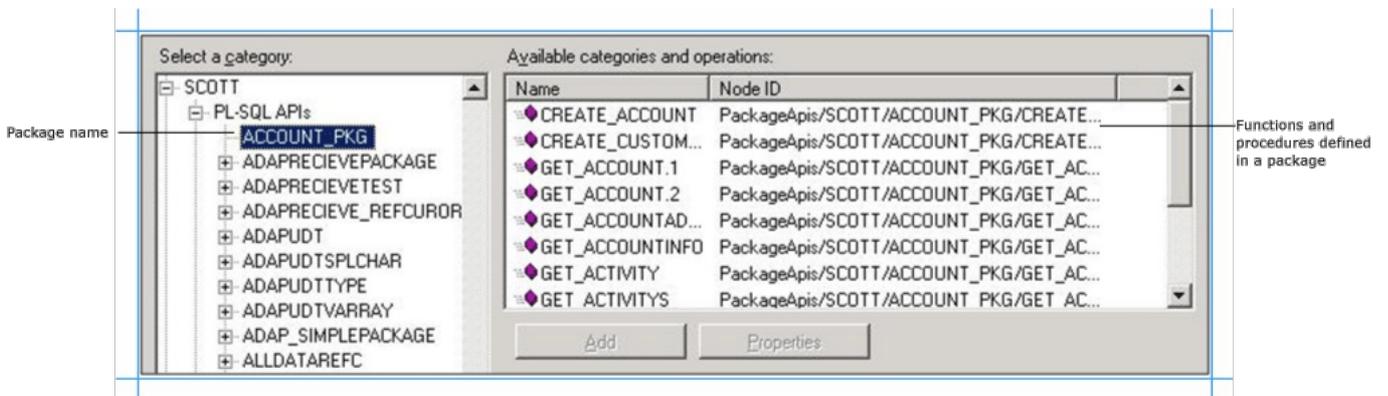
The standard operations such as ExecuteReader, ExecuteScalar, and ExecuteNonQuery are available at the root level. For more information about these operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#). For instructions on how to execute these operations using the Oracle E-Business adapter, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server](#).

4. Expand the **Schema-Based View** node to see the schemas defined in the underlying database. Each schema is further categorized on the basis of PL-SQL APIs, procedures, functions, tables, and views it contains.

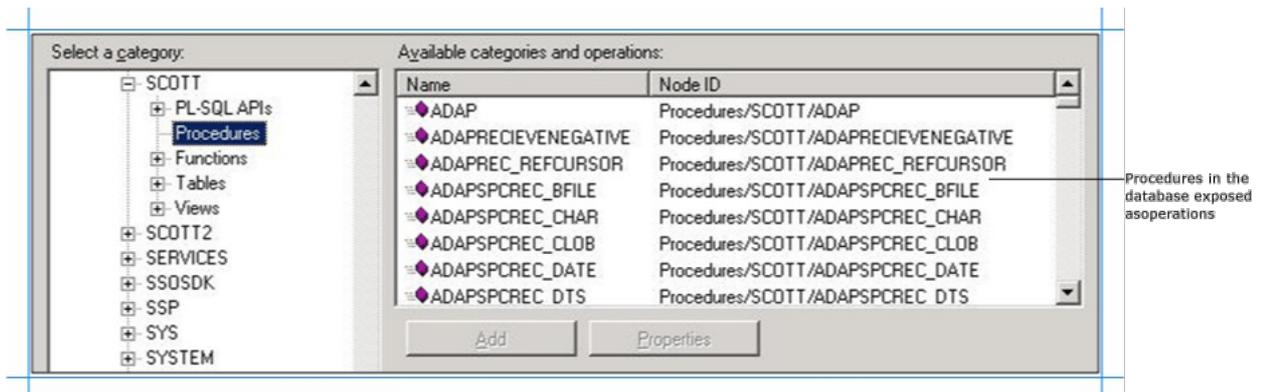
Tip

You can directly go to the "immediate" category node or subcategory nodes in the tree, by typing the name of the artifact while the focus is on the tree view in the **Select a category** box. For example, to jump to the **SCOTT** schema, keep the focus on the **Schema-Based View** node, and then type **SCOTT**.

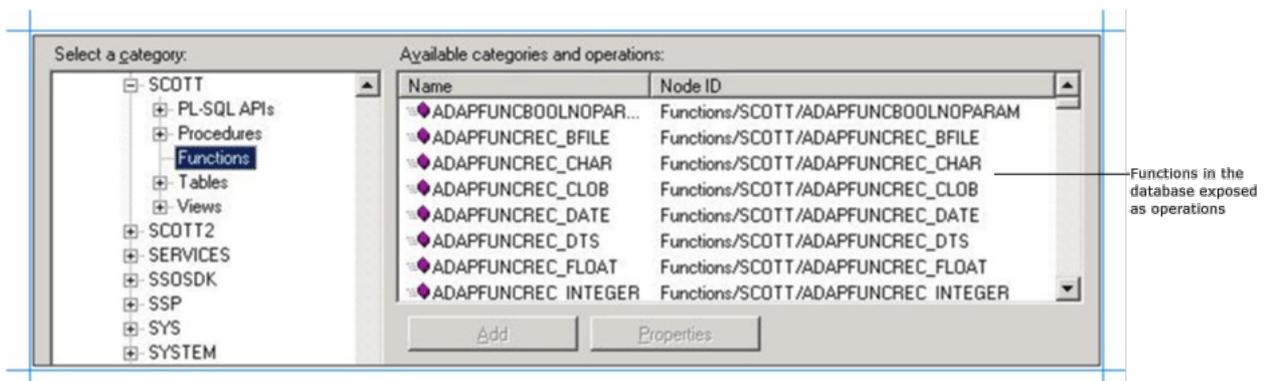
5. Expand the **PL-SQL APIs** node to see the list of packages defined in the Oracle database for a particular schema. Click a package name to see the functions and procedures defined within that package in the **Available categories and operations** box.



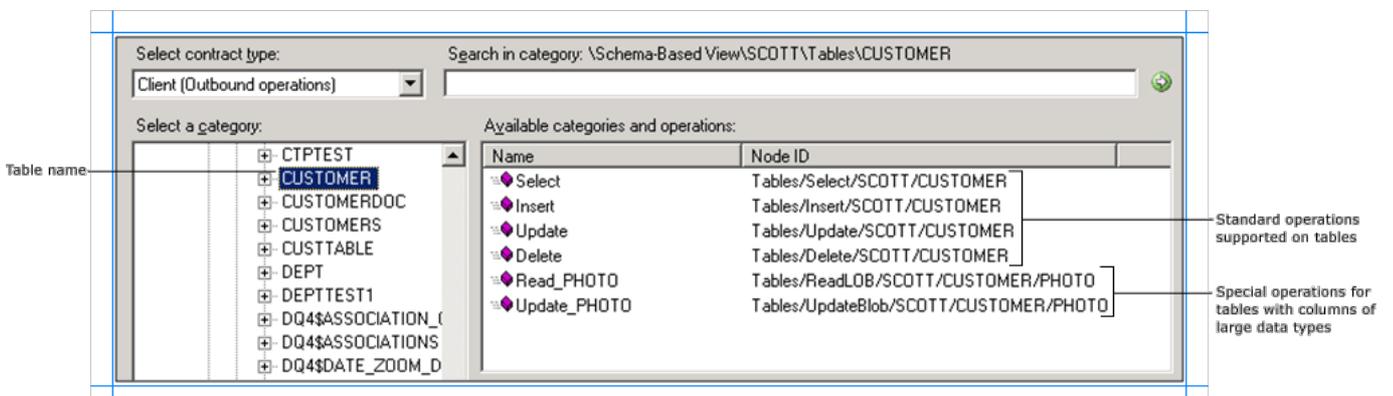
6. Click the **Procedures** node to see the list of procedures in the **Available categories and operations** box.



7. Click the **Functions** node to see the list of functions in the **Available categories and operations** box.



8. Expand the **Tables** node to see the list of tables for a particular schema. Click a table name to see the operations supported on the table in the **Available categories and operations** box.

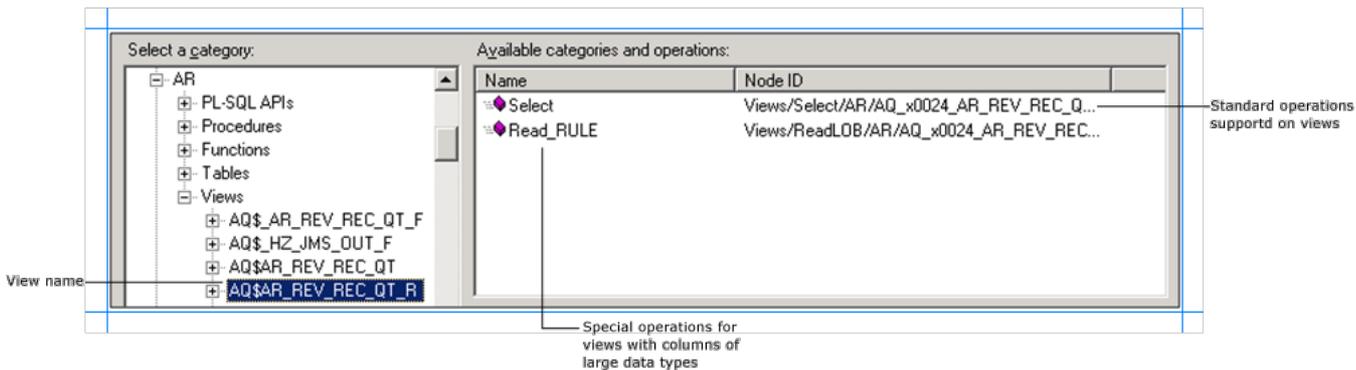


Note

If a table contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the table has a column, PHOTO, of type BLOB, the adapter exposes a **Read_PHOTO** operation. If a table has more than one column of type BLOB, CLOB, NCLOB, and BFILE the adapter will expose as many number of Read_<LOBColumnName> operations.

Similarly, if a table contains columns of type BLOB, CLOB, or NCLOB the adapter also exposes a specific operation to update data into such columns. The name of such operations are Update_<LOBColumnName>. For example, if the table has a column, PHOTO, of type BLOB, the adapter exposes an **Update_PHOTO** operation. If a table has more than one column of type BLOB, CLOB, and NCLOB the adapter will expose as many number of Update_<LOBColumnName> operations. Note that the update operation is not supported on columns of type BFILE.

- Expand the **Views** node to see the list of views for a particular schema. Click a view name to see the operations supported on the view in the **Available categories and operations** box.



Note
 If a view contains columns of type BLOB, CLOB, NCLOB, or BFILE the adapter also exposes a specific operation to read data from such columns. The name of such operations are Read_<LOBColumnName>. For example, if the view has a column, RULE, of type BLOB, the adapter exposes a **Read_RULE** operation. If a view has more than one column of type BLOB, CLOB, NCLOB, or BFILE the adapter will expose as many number of Read_<LOBColumnName> operations. Note that Update_<LOBColumnName> operations are not supported on views.

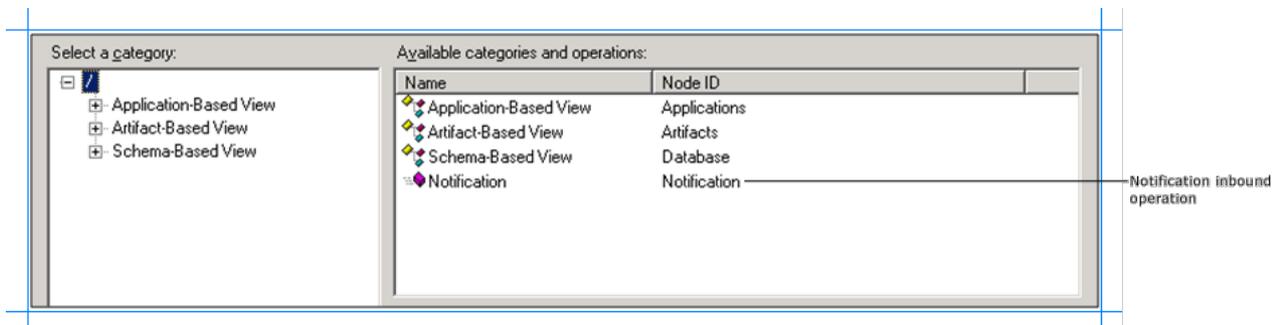
Browsing for Inbound Operations

Perform the following steps to browse the inbound operations under the schema-based view.

To browse metadata for inbound operations under the Schema-based view

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for inbound operations select **Service (Inbound operations)**.
3. The **Select a category** box lists the different views under which the Oracle E-Business Suite artifacts are categorized.

The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.

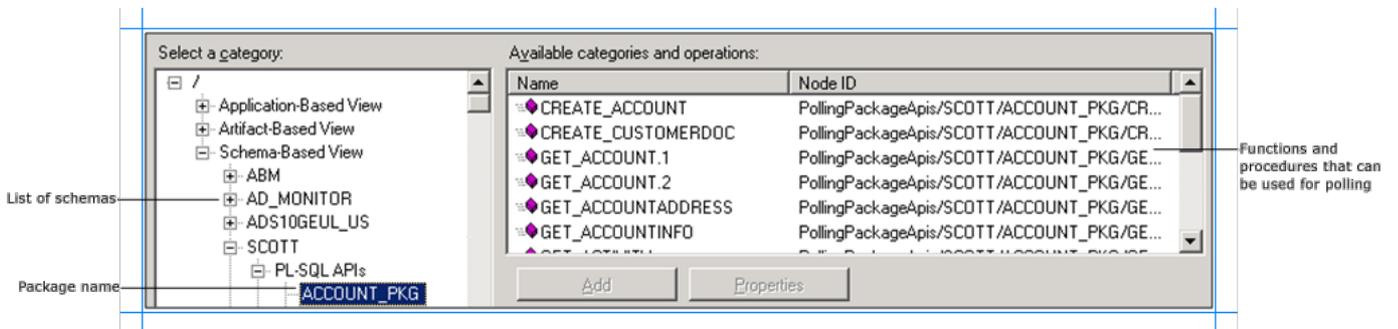


The inbound operation, **Notification**, is also available at the root level.

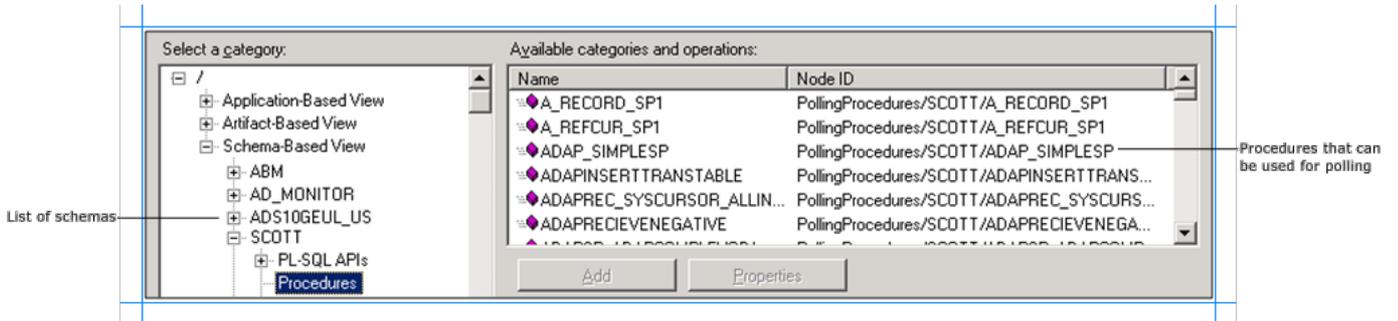
4. Expand the **Schema-Based View** node to see the schemas defined in the underlying database. Each schema is further categorized on the basis of PL-SQL APIs, procedures, functions, tables, and views it contains.

Tip
 You can directly go to the "immediate" category node or subcategory nodes in the tree, by typing the name of the artifact while the focus is on the tree view in the **Select a category** box. For example, to jump to the **SCOTT** schema, keep the focus on the **Schema-Based View** node, and then type **SCOTT**.

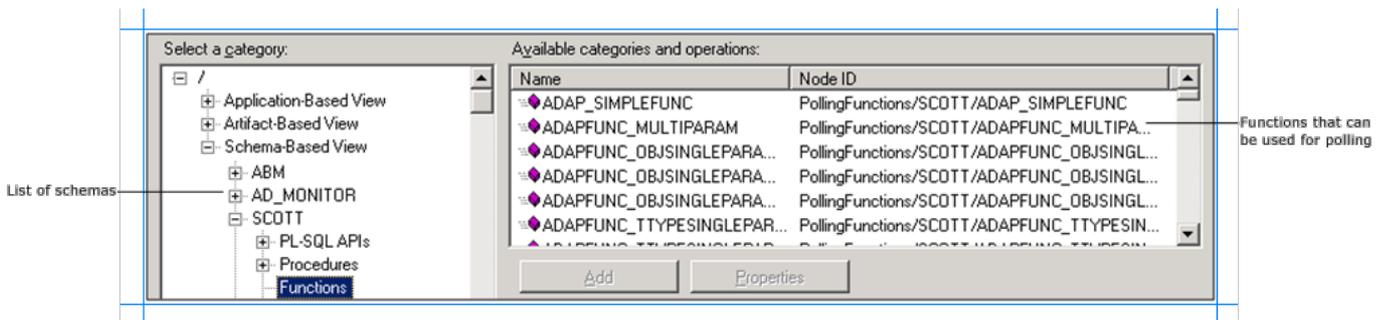
5. Expand the **PL-SQL APIs** node to see the list of packages defined in the Oracle database for a particular schema. Click a package name to see the functions and procedures defined within that package in the **Available categories and operations** box. Each of the listed functions and procedures can be used to poll the Oracle database.



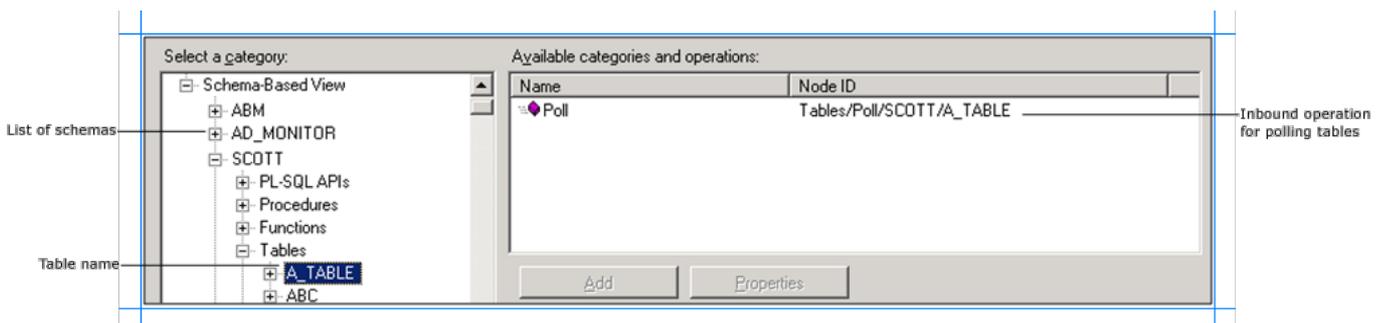
6. Click the **Procedures** node to see the list of procedures in the **Available categories and operations** box. Each of the listed procedures can be used to poll the Oracle database.



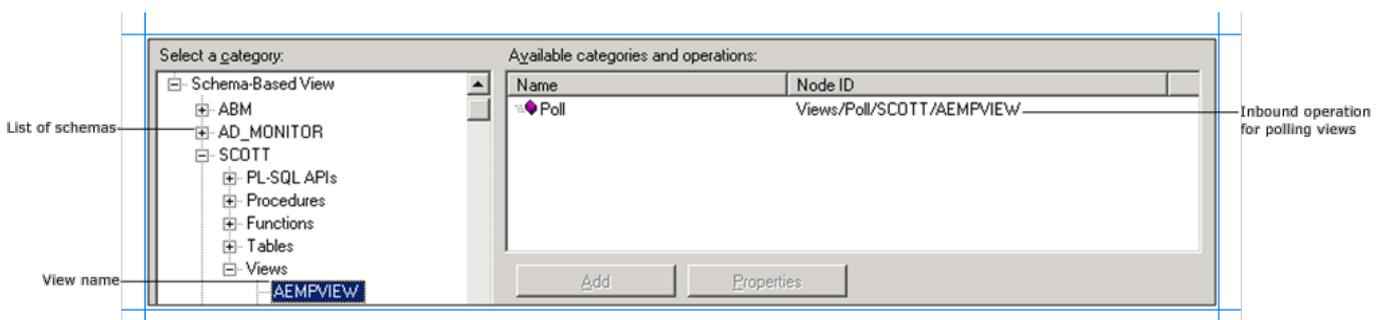
7. Click the **Functions** node to see the list of functions in the **Available categories and operations** box. Each of the listed functions can be used to poll the Oracle database.



8. Expand the **Tables** node to see the list of tables for a particular schema. Click a table name to see the **Poll** inbound operation supported on the table in the **Available categories and operations** box.



9. Expand the **Views** node to see the list of views for a particular schema. Click a view name to see the **Poll** inbound operation supported on the view in the **Available categories and operations** box.



See Also

Other Resources

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Searching for Operations

You can use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in to search for specific artifacts in the Oracle E-Business Suite. This topic provides information on how the search is supported for different views and what wildcard characters can be used for searching Oracle artifacts. This topic also provides information on how to search using the Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in.

Note

The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so both components are covered in the same topics.

For more information see [How Does the Adapter Surface Oracle E-Business Suite Metadata?](#).

Prerequisites

You must connect to the Oracle E-Business Suite before you can search metadata for target operations. For information about how to connect to the Oracle database when you use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

Support for Wildcard Characters

When searching Oracle E-Business Suite metadata using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, the Oracle E-Business adapter supports wildcard and escape characters in the search expression.

Special character	Interpretation
_ (underscore)	Matches exactly one character For example, A_ matches AB, AC, AD.
% (percentage)	Matches zero or more characters. For example, A% matches A, AB, ABC.
\ (escape)	Escapes the special meaning of % and _ For example, A_B matches A_B.

Note

Escape character is a character that is put before a wildcard character to indicate that the wildcard should be interpreted as a regular character and not as a wildcard.

Searching Under Different Views

The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in categorizes data into three views—application-based view, artifact-based view, and schema-based view. One of the reasons to categorize the artifacts under these three views is to facilitate search based on what you are searching for. The following table describes how search can differ across these views.

View	How to search
Application-based view	This view is categorized by the Oracle E-Business Suite application names. You must use this view when you know which application contains the artifacts you want to work with. Users using this view will be familiar with the Oracle applications and are aware what application they want to use. Hence, the search under this view is supported only at the immediate level. For example, if the Application-Based View node is selected, users can search for an application in the Oracle E-Business Suite. Similarly, if the Interface Table node is selected, users can search for an interface table in an Oracle E-Business application.

Artifact-based view	<p>This view is categorized by the Oracle E-Business Suite artifacts. When working with Oracle E-Business Suite application artifacts, users must use this view when they know which Oracle E-Business Suite artifact they want to work with but are not sure which application the artifact belongs to.</p> <p>Using this view, users can search for a specific artifact across all Oracle E-Business Suite applications. For example, users can select the Interface Tables node and search using the string AR%. This is how search will be performed:</p> <ul style="list-style-type: none"> • Because the interface tables are further categorized under applications, all the applications start with AR will be listed. • All the interface tables starting with AR will be listed. These tables can belong to any Oracle E-Business suite application. <p>When working with Oracle database artifacts using this view, users can search for a specific artifact either under the current schema with which you log in or all schemas. For example, if users want to use a procedure CREATE_ACCOUNT but are not aware which schema the procedure belongs to, they can select the All Schemas node and then search using the string CREATE%.</p>
Schema-based view	<p>This view is categorized by the schemas available in the underlying Oracle database. You must use this view when you know which schema has the artifacts you want to work with.</p> <p>Users using this view will be familiar with the schema that has the artifact they want to work with. Hence, the search under this view is supported only at the immediate level. For example, if the Schema-Based View node is selected, users can search for a schema in the Oracle database. Similarly, if the Table node is selected, users can search for a table in an Oracle E-Business application.</p>

Searching for Operations

To search metadata in Oracle E-Business Suite using the Consume Adapter Service Add-in, perform the following steps.

To search metadata in Oracle E-Business Suite

1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.
2. From the **Select contract type** list, select the type of contract based on whether you are searching for inbound or outbound operations.
3. In the **Select a category** box, click the category node under which you want to search for a specific artifact. For example, to search for an Oracle application, click the **Application-Based View** node.

Note

To search for an application you can specify the friendly name or the short name of the application. For example, to search for the **Receivables** application you can specify the search string as either **Receive%** or **AR**. AR is the application short name.

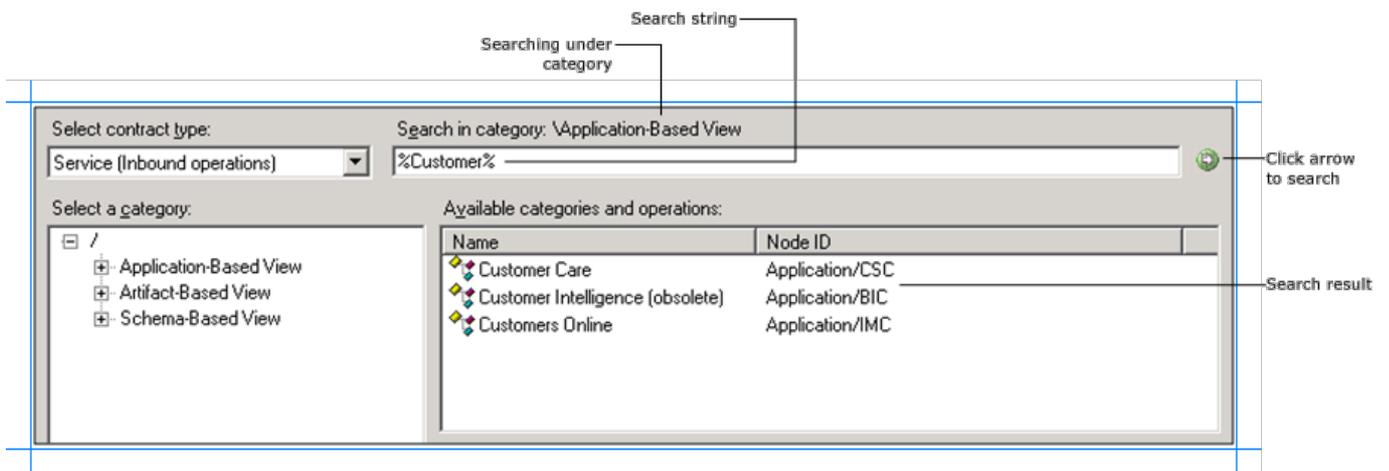
4. In the **Search in category** box, type a search expression to search for a specific artifact. For example, to search for Oracle applications that has "Customer" in their name, type **%Customer%**.

Note

The search string is case-sensitive.

5. To start the search, click the button with the right-arrow icon. After the search is complete, the **Available categories and operations** box lists the artifacts that satisfy the search criteria.

The following figure shows the Oracle applications tables that contain "Customer" in their name.



Note

To search for a concurrent program you can specify the friendly name or the actual name of the concurrent program. For example, to search for the **Customer Interface** concurrent program you can specify the search string as either **%Customer Interface%** or **%RACUST%**. RACUST is the actual name of the concurrent program.

Also, the search result will always contain the standard concurrent programs irrespective of whether their name matches with the specified search string.

See Also

Other Resources

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Retrieving Metadata for Operations

You can use the Consume Adapter Service Add-in to generate schema for selected Oracle E-Business Suite artifacts. After you have browsed and searched for the artifacts you want to invoke, you can generate schema for those artifacts and send messages, conforming to the schema, to Oracle E-Business Suite.

Note

The Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so both components are covered in the same topics.

Prerequisites

You must connect to the Oracle E-Business Suite before you retrieve metadata for target operations. For information about how to connect to the Oracle database when you use the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).

Generating Schema Using the Consume Adapter Service Add-in

Note

You can select category nodes to return all the operations in that category's sub-tree—for example, you can select the **Concurrent Programs** node (to generate schema for all the concurrent programs for an Oracle application) or you can select a specific concurrent program. For more information about the nodes, see Metadata Node IDs.

To generate schema for Oracle E-Business Suite artifacts

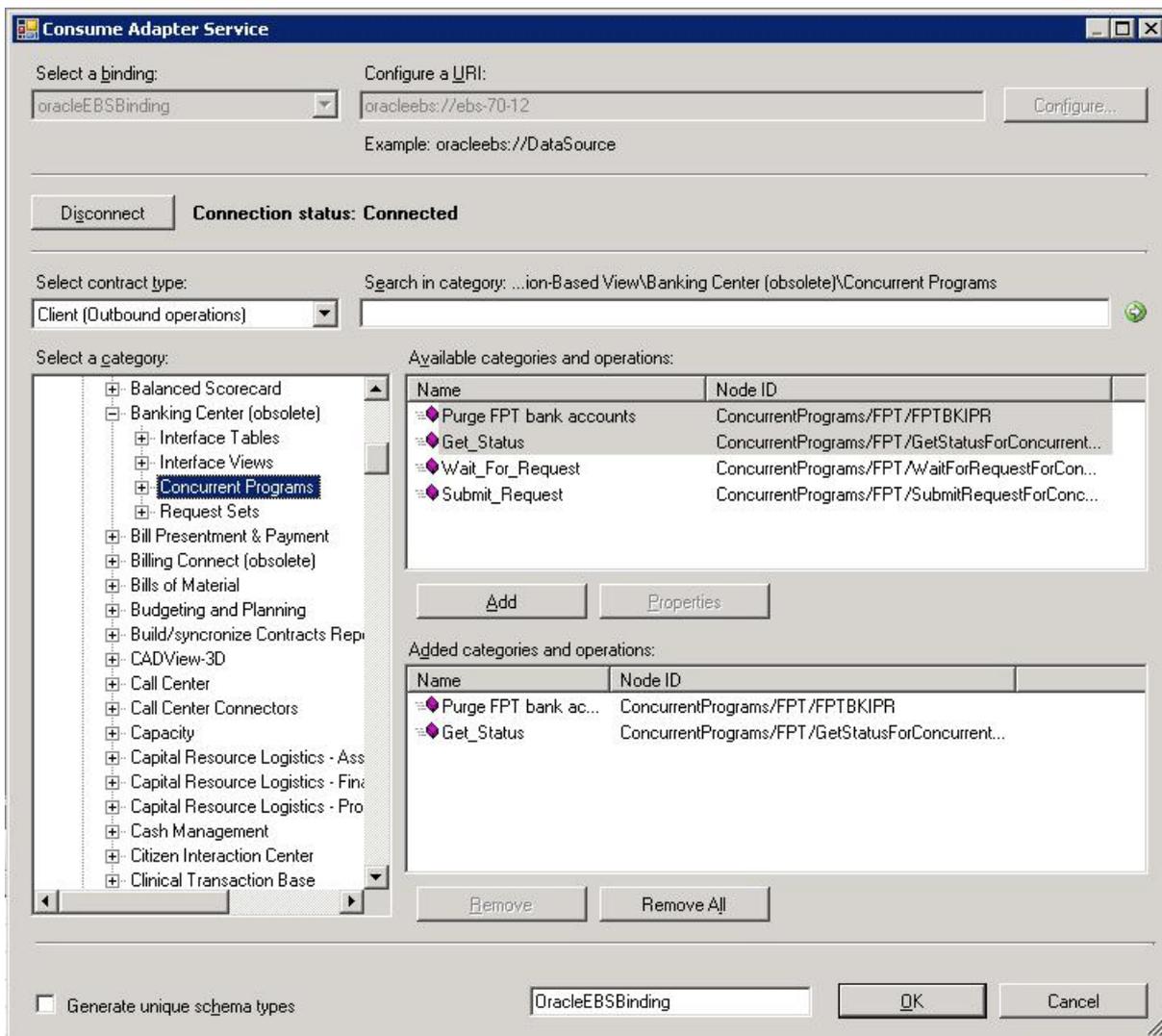
1. Connect to Oracle E-Business Suite using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. See [Connecting to the Oracle E-Business Suite in Visual Studio](#) for instructions.

Important

To generate schema for performing operations using BizTalk Server you must set the **EnableBizTalkCompatibilityMode** binding property to **True**. You must set this binding property while establishing a connection to the Oracle E-Business Suite.

2. From the **Select contract type** list, select the type of contract based on whether you are generating schema for inbound or outbound operations.
3. Click the category node for which you want to generate metadata. For example, if you want to generate metadata for a concurrent program within an Oracle application, click **Concurrent Programs**.
4. Expand the category node, and select the specific item within that node for which you want to generate metadata. For example, to generate metadata for concurrent programs for the "Banking Center" application, expand the **Concurrent Programs** node, and then click **Banking Center**.
5. In the **Available categories and operations** box, select the operations that you want to invoke and then click **Add**. The selected operations are listed in the **Added categories and operations** box. For example, to invoke "Purge FTP bank account" and "Get_Status" concurrent programs, click the operation names, and then click **Add**.

The following figure shows the Consume Adapter Service Add-in, which lists the selected operations.



If you want to generate schema for multiple operations, there may be some duplicate element definitions among these schema that may cause failure in compiling the BizTalk project. For example, consider a scenario where you generate schema for an operation "Op1". The schema for "Op1" contains a parameter of data type "CT1". After generating the schema for "Op1" you close the Consume Adapter Service Add-in and re-open it to generate schema for another operation "Op2". Assume that "Op2" also contains a parameter of data type "CT1". After you exit the Consume Adapter Service Add-in and compile the project, you will get compilation errors because the complex data type "CT1" is defined twice in different XSD files. In such situations, we recommend the following:

- Generate schema for all the operations in a single run of Consume Adapter Service Add-in. This ensures that the Consume Adapter Service Add-in generates only one definition for the complex data type "CT1".
- If you want to generate schema for multiple operations across different runs of Consume Adapter Service Add-in, make sure you select the **Generate unique schema types** check box, so that the generated XSD files contain unique namespaces for the complex data type "CT1".

6. Click **OK**. The schema file is saved with an .xsd extension at the same location as the BizTalk project.

Note

If you are using the Consume Adapter Service Add-in to generate metadata for operations on Oracle artifacts, by default the files are created with a specific naming convention: The generated XSD file name has the following three parts:

- "OracleEBSBinding" or the prefix provided in the **Filename prefix** box.
- The name included in the **fileNameHint** annotation tag in the generated WSDL. For operations, the file name hint is the same as the operation group. For the complex types, the file name hint is the namespace without the "http://schemas.microsoft.com/OracleEBS/2008/05/" prefix. For example, the file name hint for an interface table operation follows the convention <InterfaceTables> + <app_short_name> + <interface_table_name>.
- (Optional) An integer to ensure that the file name is unique.

Finally, the name of an XSD file is arrived upon as <file_name_prefix> + <fileNameHint> + n, where "n" is a unique integer.

Note

The Consume Adapter Service Add-in also creates a binding file (an XML file) containing the binding properties that you specified when generating the schema for an operation and the SOAP action to invoke the operation. You can import this binding file in the BizTalk Server Administration console to create a WCF-Custom port with the connection URI, binding properties, and the SOAP action set. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

You have successfully generated metadata for Oracle E-Business Suite artifacts. You can use the metadata to send messages to Oracle E-Business Suite to perform specific operations. See [Developing BizTalk Applications](#) for more information about how to perform these operations.

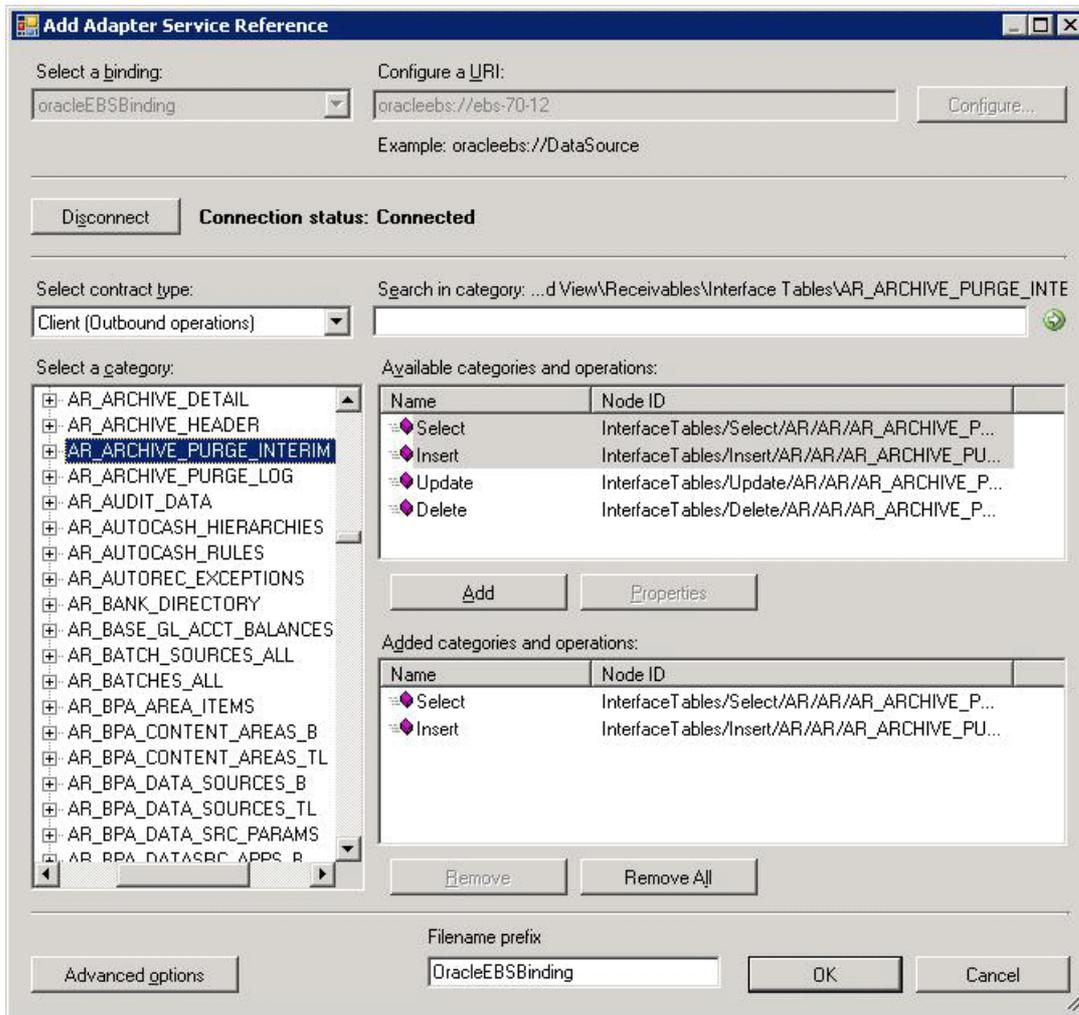
Generating a WCF Client or WCF Service Contract Using the Add Adapter Service Reference Plug-in

You can use the Add Adapter Service Reference Plug-in to generate either WCF client code for outbound operations or WCF service code for the inbound operations.

To generate WCF client class or service contract for Oracle E-Business Suite operations

1. In the Add Adapter Service Reference Plug-in, from the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound or outbound operations.
2. Browse or search for categories (such as an interface table) or for the specific operations for which you want to generate a WCF client (or WCF service contract).
For example, to browse for the operations on the AR_ARCHIVE_PURGE_INTERIM interface table, in the **Select a category** box:
 - a. Expand the root node (*J*) to see the categories under which the operations are surfaced for the Oracle E-Business Suite. The same interface table can be available under the **Application-Based View** node as well as the **Artifact-Based View** node. In this example, you generate the WCF client class from the **Application-Based View** node.
 - b. Under the root node, expand the **Application-Based View** node to see the available applications in the Oracle E-Business Suite.
 - c. Expand the node for the **Receivables** application and then expand the **Interface Tables** node.
 - d. Click the **AR_ARCHIVE_PURGE_INTERIM** interface table node, and in the **Available categories and operations** box, select the operations for which you want to generate a WCF client (or WCF service contract), and then click **Add**. The selected operations are listed in the **Added categories and operations** box.

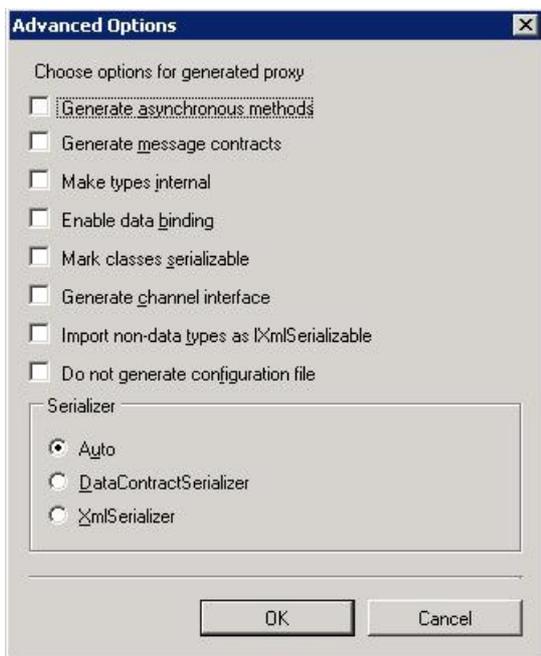
The following figure shows the Add Adapter Service Reference Plug-in with the Insert and Select operations for the AR_ARCHIVE_PURGE_INTERIM table selected.



◆ Important
 Depending on the outbound operations (or categories) that you select, more than one WCF client class may be generated. For more details, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

3. For most scenarios the default serialization options are sufficient; however, if needed, you can control several aspects about the code that is generated and the type of serializer that is used. To set these options:
 - a. Click **Advanced Options** to open the **Advanced Options** box.
 - b. In the **Advanced Options** box under **Choose options for generated proxy**, select the options that you want. For example, you can select whether asynchronous methods are generated for the WCF client or disable the generation of a configuration file.
 - c. Under **Serializer** select the serializer that should be used.

The following figure shows the **Advanced Options** box with the default selections (**Auto** is selected for the serializer and no other options are selected).



The options that you can configure in the **Advanced Options** box are equivalent to some of the options available when you use the ServiceModel Metadata Utility Tool (svcutil.exe). For more information about these options, see "ServiceModel Metadata Utility Tool (Svcutil.exe)" at <http://go.microsoft.com/fwlink/?LinkId=99821>.

4. Click **OK**. The Add Adapter Service Reference Plug-in saves the WCF client class (or WCF service interface) and helper code for the operations and categories that you have selected in your project directory. By default, a configuration file is also saved. Slightly different files are generated for inbound and outbound operations; for more information, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

You can select any node that is listed in the **Available categories and operations** box. If you select a category node then all of the operations available under that node and its sub-nodes will be selected. For example, to generate a WCF client for all of the operations surfaced for the AR_ARCHIVE_PURGE_INTERIM table, you can select the **AR_ARCHIVE_PURGE_INTERIM** node.

See Also

Other Resources

[Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#)

Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces several binding properties. By setting these properties, you can control some of the adapter's behavior. This section describes the Oracle E-Business adapter binding properties. It also shows how you can access them by using .NET programming or by setting properties on a BizTalk Server physical port binding.

The Adapter Binding Properties

The Oracle E-Business Suite is an application layer over the underlying Oracle database. Some binding properties configure the adapter's behavior while interfacing with Oracle E-Business Suite, while other properties configure the behavior while interfacing with the Oracle database. The following table shows all the Oracle E-Business adapter binding properties grouped by category. The category refers to the node under which each binding property appears in the dialog boxes that are presented by different applications to configure the adapter (or binding).

Binding Property	Category	Description	.NET Type
ClientCredentialType	General	<p>Specifies which set of credentials are specified while establishing a connection with the Oracle E-Business Suite. The possible values are Database and EBusiness.</p> <ul style="list-style-type: none"> If set to Database, adapter clients must specify the database credentials to establish the connection and the E-Business Suite credentials for the OracleUserName and OraclePassword binding properties. If set to EBusiness, adapter clients must specify the E-Business Suite credentials to establish the connection and the database credentials for the OracleUserName and OraclePassword binding properties. <p>◆ Important If ClientCredentialType binding property is set to EBusiness, specifying the database user name and password for OracleUserName and OraclePassword binding properties is mandatory.</p> <p>Default is Database.</p>	enum (ClientCredentialType)
CloseTimeout	General	The WCF connection close timeout. Specifies a time span value that indicates the interval of time provided for the internal WCF channel close operation to complete. The default is 1 minute. Not supported.	System.TimeSpan

EnableBizTalkCompatibilityMode	General	<p>Specifies how the adapter generates schema for a DataSet. Set this to True to generate the schema for DataSet correctly.</p> <p>When using the adapters from BizTalk Server, you must always set the property to True. When using the adapters from Visual Studio, you must always set the property to False.</p>	boolean (System.Boolean)
InboundOperationType	General	<p>Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling.</p> <p>For more information about Polling see Support for Inbound Calls Using Polling. For more information about Notification, see Receiving Database Change Notifications.</p>	enum
Name	General	<p>A read-only value that returns the name of the file generated by the Add Adapter Service Reference Visual Studio Plug-in to hold the WCF client class. The Add Adapter Service Reference Plug-in forms the file name by appending "Client" to the value of the Name property. The value returned is "OracleEBSBinding"; for this value, the generated file will be named "OracleEBSBindingClient".</p>	string
OpenTimeout	General	<p>The WCF connection open timeout. Specifies a time span value that indicates the interval of time provided for the internal channel open operation to complete. The default is 1 minute.</p> <p>◆Important The Oracle E-Business adapter always uses OpenTimeout to set the connection open timeout when it opens a connection to the Oracle database. The adapter ignores any timeout (System.TimeSpan) parameters passed when you open a communication object, such as a channel.</p>	System.TimeSpan
ReceiveTimeout	General	<p>The WCF message receive timeout. Specifies a time span value that indicates the maximum interval of time for which the adapter waits for an inbound message. The default is 10 minutes.</p> <p>◆Important For inbound operations such as polling, we recommend setting the timeout to the maximum possible value, which is 24.20:31:23.6470000 (24 days). When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.</p>	System.TimeSpan
SendTimeout	General	<p>The WCF message send timeout. Specifies a time span value that indicates the interval of time provided for the internal channel send operation to complete. The default is 5 minute.</p>	System.TimeSpan

Use Ambient Transaction	General	<p>Specifies whether the Oracle E-Business adapter performs the operations using the transaction context provided by the caller. Default is true, which means that the adapter always performs the operations in a transaction context, assuming that the client is providing the transactional context. If there are other resources participating in the transaction, the connections created enlist in System.Transaction and are elevated to an MSDTC transaction.</p> <p>However, there can be scenarios where you do not want the adapter to perform operations in a transactional context. For example:</p> <ul style="list-style-type: none"> • While performing a simple SELECT operation on the Oracle database (on a send port) • While specify a polling statement that performs a SELECT operation and does not involve any changes to the table either through a DELETE statement or by invoking a stored procedure (on a receive port) <p>Both these operations do not make any updates to the database table and hence, elevating these operations to use an MSDTC transaction can be a performance overhead. In such scenarios, you can set the binding property to false so that the Oracle E-Business adapter does not perform the operations in a transaction context.</p> <p> Note Not performing operations in a transactional context is advisable only for operations that do not make changes to the database. For operations that update data in the database, we recommend setting the binding property to true otherwise you might either experience message loss or duplicate messages depending on whether you are performing inbound or outbound operations.</p>	bo ol (S ys te m. Bo ol ea n)
Data Fetch Size	Buffer management	<p>ODP.NET property. Specifies the amount of data in bytes that ODP.NET fetches from the result set in one server roundtrip. The default is 65536. This property is used for performance tuning.</p>	lo ng (S ys te m. Int 64)
InsertBatchSize	Buffer management	<p>Specifies the batch size for multiple record Insert operations. The default is 20. For values of InsertBatchSize greater than one, the Oracle E-Business adapter batches the specified number of records into a single ODP.NET call. If the number of records in the Insert operation is not a multiple of the batch size, the final batch will contain fewer records than the batch size value. For example, if the insert message has 10 records and the InsertBatchSize is set to 1, the adapter reads individual records and writes them into the Oracle database. So, the adapter performs 10 separate operations on the Oracle database. Similarly, if the insert message has 10 records and the InsertBatchSize is set to 5, the adapter will read and write 5 records at a time into the Oracle database, therefore performing only 2 insert operations.</p> <p>A well-chosen value for InsertBatchSize can greatly improve adapter performance for multiple record Insert operations.</p>	int (S ys te m. Int 32)

Long Data type Column Size	Buffer management	<p>Specifies the maximum size in bytes of an Oracle long data type column. The maximum value of this binding property can be 32512. The default is 0. You must use the default value if you are not performing operation on long data type. You must explicitly set an appropriate value for this binding property if you are:</p> <ul style="list-style-type: none"> • Executing a stored procedure that contains parameters of long data type. • Performing a Select operation on a table that contains columns with long data type, and the SELECT statement does not include the primary key column. <p>If you want to set the maximum value for this binding property, set the value to -1.</p> <p> Note This binding property is deprecated.</p>	int
Max Output Associative Array Elements	Buffer management	<p>Specifies the size of the associative array that the adapter creates when performing operations that return an associative array in the response. The adapter communicates the size of the array to ODP.NET, which in turn creates a buffer depending on the array size. Default is 32.</p> <p>This binding property is useful when performing operations involving PL/SQL table types.</p>	int (System.Int32)
Metadata Poling	Buffer management	<p>ODP.NET property. Specifies whether ODP.NET caches metadata information for executed queries. The default is true, which enables metadata pooling. Caching this information improves performance; however, if changes to the underlying Oracle artifacts occur on the Oracle system, this pooled metadata will be out of sync. This might cause operations performed on the Oracle system to return unexpected exceptions. This property is used for performance tuning.</p>	boolean (System.Boolean)
StatementCachePurge	Buffer management	<p>ODP.NET property. Specifies whether the ODP.NET statement cache associated with a connection is purged when the connection is returned to the connection pool. The default is false, which disables statement cache purging. This property is used for performance tuning.</p>	boolean (System.Boolean)
StatementCacheSize	Buffer management	<p>ODP.NET property. Specifies the maximum number of statements that can be cached by each ODP.NET connection. Setting this property to a non-zero value enables statement caching for connections. The default is 10. This property is used for performance tuning.</p>	int (System.Int32)

OracleConnectionClientId	Debug	Specifies a client identifier (ID) for the connection used by the adapter to connect to Oracle. This is useful if you have enabled Oracle server-side tracing, and want to selectively trace the operations for a particular client ID. This also allows you to filter and view the Oracle server traces based on a client ID.	String
EnablePerformanceCounters	Diagnostics	Specifies whether to enable the WCF LOB Adapter SDK performance counters and the Oracle E-Business adapter LOB Latency performance counter. The default is False ; performance counters are disabled. The LOB Latency performance counter measures the total time spent by the Oracle E-Business adapter in making calls to Oracle E-Business Suite.	boolean (System.Boolean)
EnableSafeTyping	Metadata	Enables or disables safe typing. The default is false ; safe typing is disabled. This feature controls how the adapter surfaces certain Oracle data types. For more information about safe typing, see Basic Oracle Data Types .	boolean (System.Boolean)
UseSchemaNamespace	Metadata	<p>Specifies whether the schema name (SCOTT, HR, and so on) is included in the xml namespace for operations and their associated types on the underlying database artifacts. The default is True; the schema name is included in the namespace. The advantage of not having schema name included in the namespace is that if there is a table with same name (for example, EMP) in two different schemas then the same XML can be used to perform the simple SQL operations (Insert, Update, Delete, Select) on both tables.</p> <p>For example, if the UseSchemaNamespace property is true, the namespace for these operations on the SCOTT.EMP table is "http://schemas.microsoft.com/OracleEBS/Tables/SCOTT/EMP"; if it is false, the namespace is "http://schemas.microsoft.com/OracleEBS/Tables/EMP".</p> <p>◆ Important The message action is not affected by the UseSchemaNamespace binding property; it always includes the schema name.</p> <p>◆ Important We strongly recommend setting this binding property to True while generating metadata. If you set this property to false, the Oracle schema names (for example, SCOTT) will not be available in the XML namespace of the generated schema. So, if there are two tables with the same name in two different Oracle schemas, and they are added to the same BizTalk project, the BizTalk project will fail to build and deploy. If you want to include such schemas in the same BizTalk project, you must manually edit them to include the Oracle schema name in the XML namespace.</p>	boolean (System.Boolean)
MlsSettings	Multi Language Support	<p>An Oracle E-Business Suite instance can have multiple language packs installed. So, if the adapter is retrieving metadata from an Oracle E-Business Suite instance that has multi-language support (MLS), the adapter must present information with the same language setting as the logged in user prefers.</p> <p>The MlsSettings binding property enables adapter clients to specify the MLS-specific options. This binding property can be specified both at design time and run time. The MlsSettings binding property is a complex binding property that in turn contains more binding properties. All these binding properties are optional. The adapter uses the binding properties in the following manner to set the MLS context:</p>	enum

- For binding properties that have values specified, the adapter uses those values to set the MLS context.
- For binding properties that do not have values specified, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.

Following table lists the binding properties that are part of the **MlsSettings** binding property. All these properties are optional.

Property Name Description

Calendar Specifies the calendar system. This binding property maps to the NLS_CALENDAR setting in the ODP.NET client.

Comparison Specifies a method of comparison for the WHERE clauses and comparison in PL/SQL blocks. This binding property maps to the NLS_COMP setting in the ODP.NET client.

Currency Specifies the string to use as a local currency symbol for the L number format element. For information about the various number format elements and format models in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=135807>. This binding property maps to the NLS_CURRENCY setting in the ODP.NET client.

DateFormat Specifies the format in which the date is presented. This binding property maps to the NLS_DATE_FORMAT setting in the ODP.NET client.

DateLanguage Specifies the language for day and month names. This binding property maps to the NLS_DATE_LANGUAGE setting in the ODP.NET client.

DualCurrency Specifies the dual currency symbol, such as Euro, for the U number format element. For information about the various number format elements and format models in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=135807>. This binding property maps to the NLS_DUAL_CURRENCY setting in the ODP.NET client.

ISOCurrency Specifies the string to use as an international currency symbol for the C number format element. For information about the various number format elements and format models in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=135807>. This binding property maps to the NLS_ISO_CURRENCY setting in the ODP.NET client.

Language

Specifies the language to be set as part of MLS context. This binding property maps to the NLS_LANGUAGE setting in the ODP.NET client.

◆ Important

The value for the **OracleEBSResponsibilityName** binding property must be specified in the same language as specified for the **Language** binding property. For example, if you set the **Language** binding property to Spanish, the responsibility name you specify for the **OracleEBSResponsibilityName** binding property must also be in Spanish.

LengthSemantics Enables creation of CHAR and VARCHAR2 columns using either byte or character (default) length semantics. This binding property maps to the NLS_LENGTH_SEMANTICS setting in the ODP.NET client.

NCharConversionException Specifies whether the data loss during an implicit or explicit character type conversion reports an error or not. The default is **True**. This binding property maps to the NLS_NCHAR_CONV_EXCP setting in the ODP.NET client.

NumericCharacters Specifies the decimal character and group separator. This binding property maps to the NLS_NUMERIC_CHARACTERS setting in the ODP.NET client.

Sort Specifies the type of sort for character data. This binding property maps to the NLS_SORT setting in the ODP.NET client.

Territory Specifies the conventions of date and numeric format for a territory. This binding property maps to the NLS_TERRITORY setting in the ODP.NET client.

TimeStampFormat Specifies the string format for TimeStamp data types. This binding property maps to the NLS_TIMESTAMP_FORMAT setting in the ODP.NET client.

TimeStampTZFormat Specifies the string format for TimeStampTZ data types. This binding property maps to the NLS_TIMESTAMP_TZ_FORMAT setting in the ODP.NET client.

TimeZone Specifies the time zone region name. This binding property maps to the TIME_ZONE setting in the ODP.NET client.

NotificationPort	Notification	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Default is -1, which signifies that ODP.NET uses a valid, random, unused port number.</p> <p>◆Important Adapter clients will not receive database change notifications if Windows Firewall is turned on. Also, turning off Windows Firewall to receive notifications is not advisable. So, to receive notifications without compromising the security of the client-side computers, we recommend specifying a positive integer value as a port number and then adding that port number to the Windows Firewall exceptions list. If you set this binding property to the default value of -1, ODP.NET uses a random port and adapter clients will not know which port to add to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>⚠Caution If there is more than one application in an application domain receiving notifications using the Oracle E-Business adapter, the NotificationPort binding property for all applications must be set to the same port number. This is because ODP.NET creates only one listener that listens on one port within an application domain.</p>	int (System.Int32)
NotificationStatement	Notification	<p>Specifies the SELECT statement used to register for getting notifications from Oracle database. An example SELECT statement could resemble the following.</p> <pre style="border: 1px solid black; padding: 5px; margin: 10px 0;">SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌Note You must specify the database object name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p> <p>The adapter gets a notification message from Oracle database only when the result set for the specified SELECT statement changes.</p>	string
NotifyOnListenerStart	Notification	<p>Specifies whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts. Default is True.</p>	bool (System.Boolean)
ConnectionLifetime	Oracle Connection Pool	<p>ODP.NET property. Specifies the maximum duration in seconds of a connection. The default is 0. This property is used for performance tuning.</p>	int (System.Int32)

Decr Pool Size	Oracle Connection Pool	ODP.NET property. Specifies the number of connections that are closed when an excessive amount of established connections are not in use. The default is 1. This is used for performance tuning.	int (System.Int32)
Incr Pool Size	Oracle Connection Pool	ODP.NET property. Specifies the number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool. The default is 5. This property is used for performance tuning.	int (System.Int32)
Max Pool Size	Oracle Connection Pool	ODP.NET property. Specifies the maximum number of connections in an ODP.NET connection pool. The default is 100. This property is used for performance tuning. ♦Important You must set MaxPoolSize judiciously. It is possible to exhaust the number of connections available from ODP.NET, if this value is set too large.	int (System.Int32)
Min Pool Size	Oracle Connection Pool	ODP.NET property. Specifies the minimum number of connections in an ODP.NET connection pool. The default is 1. This property is used for performance tuning.	int (System.Int32)
Use OracleConnection Pool	Oracle Connection Pool	ODP.NET property. Specifies whether to use the ODP.NET connection pool. The default is true , which enables connection pooling. The Oracle E-Business adapter implements connection pooling by using the ODP.NET connection pool.	boolean (System.Boolean)

Application Short Name	OracleEBS	<p>The short name for an Oracle E-Business Suite application. You must provide the application short name while setting the application context for certain PL/SQL APIs, and database tables or views. But, why do we need to set the application context for only certain PL/SQL APIs, and database tables or views?</p> <ul style="list-style-type: none"> • The PL/SQL APIs, both associated with the Oracle database and the applications in the Oracle E-Business Suite, are also available under the "Artifact-Based View" and the Schema-Based View" node at design-time while using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. While executing the PL/SQL APIs associated with the Oracle database, you need not set this binding property. However, while executing a PL/SQL API associated with an Oracle E-Business Suite application, you must set the application context by specifying a valid value for this binding property. • For custom Oracle E-Business Suite applications, users may or may not register the base database tables as interface tables. If a database table is not registered as an interface table, it will be available under the "Database" node at design-time while using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in. However, because these tables are associated with an Oracle E-Business application, for any operation on these tables you must set the application context <p>So, to perform any operation on a PL/SQL API, and database tables or views belonging to the Oracle E-Business Suite, you must always specify the application short name.</p> <p>◆ Important While executing a PL/SQL API or performing any operation on database tables or views associated with an Oracle E-Business Suite application, you must also:</p> <ul style="list-style-type: none"> • Set the ClientCredentialType binding property to EBusiness, • Set the OracleUserName, OraclePassword binding properties to specify the credentials to connect to the Oracle E-Business Suite, and • Set the OracleEBSResponsibilityName binding property to the appropriate responsibility name <p>By setting these binding properties you inform the adapter that the PL/SQL API, or the database table or view, you are operating on is associated with an Oracle E-Business Suite application and hence the ApplicationShortName binding property becomes mandatory. On the contrary if you are executing a PL/SQL API, or any operation on database tables or views, associated with an Oracle E-Business Suite application, and you do not set the ClientCredentialType, OracleUserName, OraclePassword, and OracleEBSResponsibilityName binding properties appropriately, the adapter ignores the ApplicationShortName, even if it is set, and assumes that the PL/SQL API or the database table you are performing operations on belongs to the Oracle database.</p>	string
OracleEBSOrganizationId	OracleEBS	<p>Specifies the organization ID to which an Oracle E-Business Suite application belongs. This property is optional and is considered only when:</p> <ul style="list-style-type: none"> • The ClientCredentialType binding property is set to EBusiness, and • The OracleUserName, OraclePassword, and OracleEBSResponsibilityName binding properties are specified appropriately. 	string

OracleEBSResponsibilityKey	OracleEBS	<p>The responsibility key associated with the Oracle E-Business Suite user.</p> <p>Unlike the responsibility name (OracleEBSResponsibilityName binding property), the responsibility key is not dependent on the language specified. It implies that you do not have to specify responsibility keys in different languages depending on the value specified in the Language binding property while setting the MLS context.</p> <p>◆Important</p> <ul style="list-style-type: none"> • The value specified for the OracleEBSResponsibilityKey binding property overrides the value specified for the OracleEBSResponsibilityName binding property. • If you will be performing operations on the data in multiple languages and do not want to specify a responsibility name in each language, you should specify a value for this binding property while performing operations on interface tables, selecting values from interface views, or executing a concurrent program or request set. Of course, you can always use this binding property even if you are working only with data in the English language. 	string
OracleEBSResponsibilityName	OracleEBS	<p>The name of the responsibility associated with the Oracle E-Business Suite user.</p> <p>The responsibility name is dependent on the language specified. It implies that if you are setting the MLS context by specifying a value for the Language binding property, the value for the OracleEBSResponsibilityName binding property must be specified in the same language as specified for the Language binding property. For example, if you set the Language binding property to Spanish, the responsibility name you specify for the OracleEBSResponsibilityName binding property must also be in Spanish.</p> <p>◆Important</p> <p>You must specify a value for this binding property while performing operations on interface tables, selecting values from interface views, or executing a concurrent program.</p>	string
OraclePassword	OracleEBS	<ul style="list-style-type: none"> • If the ClientCredentialType binding property is set to Database, this property specifies the password for an Oracle database user. • If the ClientCredentialType binding property is set to EBusiness, this property specifies the password for an Oracle E-Business Suite user. <p>The Oracle E-Business adapter does not preserve the case of the value that you enter for the OraclePassword binding property when it connects to Oracle E-Business Suite. The password is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.</p> <p>◆Important</p> <ul style="list-style-type: none"> • While performing operations on interface tables, selecting values from interface views, executing a concurrent program, or executing a PL/SQL API associated with an Oracle E-Business Suite application you must set the ClientCredentialType binding property to EBusiness and then specify a valid password to connect to Oracle E-Business Suite. • The OraclePassword binding property is not available in the binding file (XML file) or the app.config file generated as a result of using the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in. 	string

OracleUserName	OracleEBS	<ul style="list-style-type: none"> If the ClientCredentialType binding property is set to Database, this property specifies the user name for authentication on the Oracle database. If the ClientCredentialType binding property is set to EBusiness, this property specifies the user name for authentication on the Oracle E-Business Suite. <p>The Oracle E-Business adapter does not preserve the case of the value that you enter for the OracleUserName binding property when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.</p> <p>◆ Important While performing operations on interface tables, selecting values from interface views, executing a concurrent program, or executing a PL/SQL API associated with an Oracle E-Business Suite application you must set the ClientCredentialType binding property to EBusiness and then specify a valid user name to connect to Oracle E-Business Suite.</p>	string
PolledDataAvailableStatement	Polling	<p>Specifies the SELECT statement executed to determine whether any data is available for polling for a specific table. The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the value specified for the PollingInput binding property. If the first cell of the result contains a positive value, the adapter executes the polling statement. For example, a valid statement for this binding property will be:</p> <div data-bbox="304 913 1460 994" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>Select * from <table_name></pre> </div> <p>📌 Tip If you want the adapter to continue polling irrespective of whether the table being polled has data, you can specify the value for this binding property as <code>Select 1 FROM DUAL</code>.</p> <p>📌 Note You must not specify stored procedures for this binding property. Also, this statement must not modify the underlying Oracle database.</p>	string
PollingAction	Polling	<p>Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.</p>	string
PollingInput	Polling	<p>Specifies the polling statement. You can specify a simple SELECT statement or a stored procedure for polling. If you want to poll a table or view, you must specify a SELECT statement for this binding property. If you want to poll using a stored procedure, you must specify the entire request message for this binding property. The request message must be the same that you send to the adapter for invoking the stored procedure as an outbound operation.</p> <p>📌 Note The SQL statement or the stored procedure specified for this binding property is executed only if the statement executed by the PolledDataAvailableStatement binding property returns some data.</p> <p>◆ Important The Oracle E-Business adapter executes the polling statement and the post-poll statement (if specified) inside of an Oracle transaction. If you are using a SELECT statement in the PollingInput binding property, we recommend that you specify a FOR UPDATE clause in your SELECT statement. This will ensure that the selected records are locked during the transaction and that the post-poll statement can perform any required updates on the selected records.</p>	string

PollingInterval	Polling	<p>Specifies the transacted polling interval, that is, the interval in seconds at which the Oracle E-Business adapter executes the statement you specified for PollingInput binding property against the Oracle database. The default is 30. The polling interval is used by the adapter for the following:</p> <ul style="list-style-type: none"> • The time interval between successive polls. This interval is used to run the poll and post-poll queries. If these queries are executed within the specified interval, the adapter sleeps for the remaining time in the interval. • The polling transaction timeout value. This value must be set large enough to include the polling statement execution time, the post-poll statement (if specified) execution time, and the time to receive the reply from the client application to commit the transaction. <p>If the client application sends a reply before the polling interval expires, the adapter commits the transaction and waits until the polling interval is reached to execute the next poll.</p> <p>If the client application returns a fault, the adapter terminates the transaction.</p> <p>If the polling interval expires before the client application sends the reply, the transaction will time out.</p>	int (S ys te m. Int 32)
PollWhileDataFound	Polling	<p>Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false.</p> <p>Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for PolledDataAvailableStatement returns that data is available for polling. The adapter then executes the statement specified for the PollingInput binding property. Assuming that the adapter takes just 10 seconds to execute the statement, it will now have to wait for 50 seconds before executing the PolledDataAvailableStatement again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the PollWhileDataFound binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.</p> <p> Note This binding property is applicable both for polling on tables and views and polling using stored procedure.</p>	str in g
PostPollStatement	Polling	<p>Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed. The default is null; no post-poll statement is executed. The post-poll statement executes inside the polling transaction. Two common uses for the post-poll statement are to:</p> <ul style="list-style-type: none"> • Update a column in the rows returned in the polling statement to indicate that they have been processed and should be excluded from subsequent polling queries. • Move processed records to a different table. <p> Important If a post-poll statement is specified, PollingInterval should be set large enough for the post-poll statement to complete before the interval expires.</p> <p>For more information about how to use binding properties in a polling scenario, see Overview of Polling-based Data-changed Notifications.</p>	str in g

SkipNilNodes	Run Time Behavior	<p>Specifies whether the Oracle E-Business adapter will skip inserting or updating values for nodes that are marked as 'nil' in the request XML. This binding property is applicable for inserting or updating records in a table and for RECORD type parameters in stored procedures. Default is true, which means the adapter will skip passing values for nodes that are marked as 'nil'. In this case, the default value in Oracle (if specified) is taken into account for nodes that are marked as 'nil'. If set to false, the adapter explicitly passes a null value for these nodes.</p> <p> Note</p> <ul style="list-style-type: none"> • For nodes that are not present in the request XML, the adapter always skips passing values, irrespective of the value of the SkipNilNodes binding property. • For PL/SQL tables of RECORDS, the adapter always passes a null value for nodes that are either marked as 'nil' or not present in the request XML, irrespective of the value of the SkipNilNodes binding property. <p>The following example explains the difference in the adapter configuration based on the value you set for this binding property. Assume a request XML resembles the following:</p> <pre data-bbox="343 795 662 896"><EMPNO>1000</EMPNO> <ENAME>John</ENAME> <SAL nil='true'></SAL></pre> <p>If SkipNilNodes is set to true, the adapter executes the following command:</p> <pre data-bbox="343 1008 1109 1041">INSERT INTO EMP (EMPNO, ENAME) VALUES (1000, "John");</pre> <p>If SkipNilNodes is set to false, the adapter executes the following query:</p> <pre data-bbox="343 1164 1268 1198">INSERT INTO EMP (EMPNO, ENAME, SAL) VALUES (1000, "John", null);</pre> <p>Note that in the second statement, the adapter explicitly inserts a null value for the parameter "SAL".</p>	bo ol (S ys te m. Bo ol ea n)
GeneratedUserTypesAssemblyPath	UDT .NET Type Generation – Design Time	<p>Specifies the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata. You must specify a DLL name if you are generating metadata for tables or packages, stored procedures, or functions that use UDTs. Specifying the DLL name is optional for tables and views that have UDTs. The generated DLL is saved to the same location as the executable.</p> <p>This binding property is required only while generating metadata.</p> <p> Note</p> <ul style="list-style-type: none"> • You must specify only one filename. For all the UDTs in the metadata, the adapter generates a single file with the given name. If you do not specify a name, the adapter generates the DLL with a GUID name. • This binding property is not available in BizTalk Server while configuring a WCF-OracleEBS receive or send port. 	str in g

GeneratedUserTypesAssemblyKeyFilePath	UDT .NET Type Generation – Design Time	<p>Specifies the name and path of the key file that the adapter uses to create a strongly-typed assembly.</p> <p>This binding property is optional and is required only while generating metadata.</p> <p> Note This binding property is not available in BizTalk Server while configuring a WCF-OracleEBS receive or send port.</p>	string
UserAssemblyLoadPath	UDT .NET Type Generation – Run Time	<p>Specifies the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata. These DLLs are saved at the location you specified for the GeneratedUserTypesAssemblyKeyFilePath binding property while generating metadata. You must manually copy these DLLs to the following locations:</p> <ul style="list-style-type: none"> • For BizTalk projects: Copy the DLLs at the same location as BTSNTSvc.exe. For BizTalk Server 2010, this is available typically under <installation drive>\Program Files\Microsoft BizTalk Server 2010. • For .NET Projects: Copy the DLLs to the \bin\Development folder within your .NET project folder. <p>This binding property is required only while sending and receiving messages to perform operations on the Oracle E-Business Suite.</p>	string
AcceptCredentialsInUri	Not surfaced by the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in.	<p>Specifies whether the Oracle E-Business Suite connection URI can contain user credentials. The default is false, which disables user credentials in the connection URI. If AcceptCredentialsInUri is false and the Oracle connection URI contains user credentials, the Oracle E-Business adapter throws an exception. You can set AcceptCredentialsInUri to true if you must specify credentials in the URI. For more information, see The Oracle E-Business Suite Connection URI.</p>	boolean

How Do I Set the Binding Properties?

You can set the binding properties when you specify a connection to the Oracle E-Business Suite. For information about how to set binding properties when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to the Oracle E-Business Suite in Visual Studio](#).
- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

◆ Important

While using the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file (an XML file) or the app.config file respectively. You must manually add the binding property and its value in the binding file or the app.config file, if required.

However, the **OraclePassword** binding property is not available in the binding file or the app.config file even if you specify a value for this binding property while using the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in.

See Also

Other Resources

[Development](#)

Receiving Polling-based Data-changed Messages

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for Oracle E-Business Suite supports receiving polling-based data-changed messages by polling the interface tables, interface views, tables, and views. The adapter delivers the messages to your application by:

- Executing a SQL SELECT query to determine whether data is available for polling. You can configure the adapter to execute the SQL SELECT query periodically or continuously.
- Executing a SQL SELECT query against an Oracle table or view or executing stored procedures, functions, or packaged procedures and functions.
- Executing an optional post-poll PL/SQL code block on Oracle E-Business Suite. This code block is often used to update a field on the queried records in the target or to move the queried records to another table or view.
- Returning the query results in a result set by invoking the POLL operation or the stored procedures, functions, or packaged procedures and functions that are exposed as polling operations.

The adapter executes all of these operations inside an Oracle transaction.

How Do I Configure the Oracle E-Business Adapter for Receiving Data-changed Messages Using Binding Properties?

You configure the Oracle E-Business adapter to receive data-changed messages by setting some or all of the following binding properties.

Binding Property	Value	Default	Required/Optional
InboundOperationType	Make sure that the value is set to Polling .	Polling	Required. If not explicitly set, the default value will apply.
PollableStatement	<p>Specify the SELECT statement executed to determine whether any data is available for polling for a specific table. The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the value specified for the PollingInput binding property. If the first cell of the result contains a positive value, the adapter executes the polling statement. For example, a valid statement for this binding property will be:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre>Select * from <table_name></pre> </div> <p> Note You must not specify stored procedures for this binding property. Also, this statement must not modify the data in Oracle E-Business Suite or the underlying Oracle database.</p>	Null	Required.
PollingAction	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.	Polling	Optional for polling operations on tables and views using the SELECT statement.

PollingInput	Specify either of the following: <ul style="list-style-type: none"> SQL SELECT statement that should be executed against Oracle E-Business Suite. This statement should include a FOR UPDATE clause. For information about the FOR UPDATE clause, see Specifying a FOR UPDATE Clause in the Polling Statement later in this topic. Request message for a stored procedure, function, or procedure or function within a package that you want to be polled. 	Required. Setting PollingInput to a non-null value enables polling.
PollingInterval	Set to the interval, in seconds, at which you want the adapter to query Oracle E-Business Suite. This property specifies the polling interval and the polling transaction time out. The value should be greater than the amount of time it takes to execute the query and post-poll statement (if one is specified) on Oracle E-Business Suite plus the amount of time it takes for the client to process the query data and return the polling response message.	Required. If not explicitly set, the default value will apply.
WhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously polls Oracle E-Business Suite, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval	Required. If not explicitly set, the default value will apply.
PostPollStatement	Set to an optional PL/SQL code block that is executed by the adapter after the query is performed, but before the query data is returned to the client.	Optional. If no value is specified, a post poll statement is not executed.

Note
If you are using the WCF service model or the WCF channel model, you must also set the **AcceptCredentialsInUri** binding property.

Specifying a FOR UPDATE Clause in the Polling Statement

If you are using a SELECT statement as the polling statement and executing a post-poll statement that affects the rows specified in the SELECT statement, you must use the FOR UPDATE clause in the polling statement. Specifying a FOR UPDATE clause ensures that the records selected by the polling statement are locked during the transaction and that the post-poll statement can perform any required updates on them.

Caution
You can have scenarios where in the time window between the polling and post-poll statements, more records are added to the table that meet the condition of the post-poll statement. In such situations, the post-poll statement would update all the records that satisfy the condition and not just the records selected as part of the polling statement.

If a post-poll statement is specified and the polling statement does not contain a FOR UPDATE clause, you will experience one of the following two conditions:

- If **TransactionIsolationLevel** is set to **ReadCommitted**, the post-poll query will not update the selected rows.
- If **TransactionIsolationLevel** is set to **Serializable**, the following target system exception (**Microsoft.ServiceModel.Channels.Common.TargetSystemException**) will occur when the post-poll statement is executed: "ORA-08177 can't serialize access for this transaction". In such a case, you must set the **PollingRetryCount** binding property to define the number of times you want the adapter to retry the same transaction.

For instructions on how to set the transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).

The polling and post-poll statements are executed in a transaction if the adapter clients have configured to use transactions and the value of the **UseAmbientTransaction** binding property is set to **True** in the adapter.

An example of a polling query with the FOR UPDATE option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE
```

Specifying a NOWAIT Clause in the Polling Statement

You may have scenarios where concurrent threads are accessing the table being polled, leading to too many contentions in the table. This may cause the polling query to be blocked to get a lock on table rows. If you are using a SELECT statement as the polling statement, you may want to specify a NOWAIT keyword along with the FOR UPDATE keyword in the SELECT statement. This will cause the polling query execution within the adapter to return immediately if there are locks on rows which the polling query is trying to select. An exception is usually thrown by Oracle under such conditions. Again, adapter clients may use the **PollingInterval** binding property to specify the time interval after which the adapter clients must retry for polling the data.

An example of a polling query with the NOWAIT option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE NOWAIT
```

Specifying a SKIP LOCKED Clause in the Polling Statement

You may have scenarios where due to concurrent threads accessing the table being polled, some rows in the result set of the WHERE clause specified in the polling query are locked. For example, your polling query returns 6 rows from a table; 4 out of these 6 rows are already locked because of some other transaction. In this case, you might want to specify a SKIP LOCKED keyword along with the FOR UPDATE keyword that instructs the database to attempt to lock the rows specified by the WHERE clause, and to skip any rows that are found to be already locked. The unlocked rows in the WHERE clause are locked during the transaction and the post-poll statement can perform any required updates on them so that these rows are not polled again. This ensures that you do not have to wait to receive the polling messages until all the rows specified by the WHERE clause are unlocked.

The SKIP LOCKED keyword is useful in a scenario where you have adapter clients on multiple computers that are polling the same table in a database. You can load balance among the adapter clients by configuring the polling operation in such a way that you receive polling-based data-change messages for the rows specified by the WHERE clause that are unlocked at that point of time, and then update the row to ensure that if a polling-based data-change message is received by an adapter client, the other clients do not get the same message.

An example of a polling query with the SKIP LOCKED option is:

```
SELECT * from EMP WHERE FLAG = 'Y' FOR UPDATE SKIP LOCKED
```

Support for Ordered Delivery (FIFO)

In a production environment, polling can be used to monitor the data changes in Oracle E-Business Suite. These data-changed messages are received by the adapter client using the Oracle E-Business adapter. Based on business scenarios, it can be critical that the data-changed messages are received by the adapter client in the right order.

The Oracle E-Business adapter supports ordered delivery or first-in-first-out (FIFO) to maintain the order in which messages are received from Oracle E-Business Suite. Here are a few considerations related to support for FIFO in inbound scenarios for the Oracle E-Business adapter.

- If the message is being consumed by an orchestration, the orchestration must have the ordered delivery set for the messages coming from the Oracle E-Business adapter receive port.
- If the message is being consumed by a send port (in a content-based routing) scenario, the send port must have ordered delivery set for the messages coming from the Oracle E-Business adapter receive port.

The WCF-Custom or WCF-OracleEBS adapter has a property **Suspend request message on failure** that specifies whether to suspend the request message that fails inbound processing. This property can be set on the **Messages** tab of the WCF-Custom or WCF-OracleEBS receive port under the **Error handling** section. The following table lists the scenarios describing how the incoming messages are processed based on whether this property is set and the state of the message subscriber (orchestration or port).

WCF-Custom port property	Subscriber in Unenlisted state	Subscriber in Enlisted but Stopped state
--------------------------	--------------------------------	--

<p>Suspend request message on failure property NOT set</p>	<ul style="list-style-type: none"> • Routing Failure Report is generated as a suspended (non-resumable message) • Actual message is not suspended • Post poll query is not executed as transaction gets aborted. Hence polling repeats and fetches the rows again. • Errors reported in the event log to describe what has happened. 	<ul style="list-style-type: none"> • Not considered a "Failure". There are no error messages in the event log. • Actual message is put into the suspended (resumable) queue. • When the subscribing port or orchestration starts, the messages are automatically resumed. If ordered delivery is set on the subscriber, it will be honored. • The messages may also be resumed manually.
<p>Suspend request message on failure property IS set</p>	<ul style="list-style-type: none"> • Routing Failure Report is generated as a suspended (non-resumable message) • Actual message is also suspended • Post poll query is not executed as transaction gets aborted. Hence polling repeats and fetches the rows again. • Errors reported in the event log to describe what has happened. 	<ul style="list-style-type: none"> • Not considered a "Failure". There are no error messages in the event log. • Actual message is put into the suspended (resumable) queue. • When the subscribing port or orchestration starts, the messages are automatically resumed. If ordered delivery is set on the subscriber, it will be honored. • The messages may also be resumed manually.

See Also

Other Resources

[Development](#)

[Polling Oracle E-Business Suite Using BizTalk Server](#)

Before Developing Applications Using the Oracle E-Business Suite Adapter

This section provides information about what you must do before developing applications using the Oracle E-Business adapter.

In This Section

- [Configure MSDTC on Oracle E-Business Suite and Adapter Client](#)

Configure MSDTC on Oracle E-Business Suite and Adapter Client

The operations performed on Oracle E-Business Suite using the Oracle E-Business adapter (through BizTalk Server, the WCF service model, or the WCF channel model) can be performed within a transaction scope. If the client program has more than one transactional resource as part of the same transaction, the transaction gets elevated to an MSDTC transaction. To enable the adapter to perform operations within the scope of an MSDTC transaction, you must configure MSDTC both on the computer running the Oracle E-Business adapter and Oracle E-Business Suite. Also, you must add MSDTC to the exceptions list of Windows Firewall. This section provides information about how to perform these tasks on computers running the adapter client and Oracle E-Business Suite.

Note

The steps to configure MSDTC may vary slightly for different operating systems. The steps mentioned in this section apply to Windows Client and Windows Server Operating system.

To configure MSDTC

1. Start the Component Services MMC snap-in.
 - Click **Start**, click **Control Panel**, double-click **Administrative Tools**, and then double-click **Component Services**.
2. In the Component Services MMC snap-in, from the left pane expand **Component Services**, expand **Computers**, right-click **My Computer**, and then click **Properties**.
3. In the **My Computer Properties** dialog box, click the **MSDTC** tab.
4. In the **Transaction Configuration** section, click **Security Configuration**.
5. In the **Security Configuration** dialog box, select the **Network DTC Access** check box and within that, select the **Allow Remote Clients** check box.
6. In the **Transaction Manager Communication** section, select the **Allow Inbound** and **Allow Outbound** check boxes.
7. In the **Security Configuration** dialog box, click **OK**.
8. In the dialog box informing that the MSDTC service will be restarted, click **Yes**. After the MSDTC service is restarted, click **OK**.
9. In the **My Computer Properties** dialog box, click **OK**.

To add MSDTC to Windows Firewall exceptions list

1. Open the **Windows Firewall** dialog box.
 - Click **Start**, click **Control Panel**, and then double-click **Windows Firewall**.
2. In the **Windows Firewall** dialog box, click the **Advanced Settings** tab on the right pane,.
3. In the **Windows Firewall with Advanced Settings** Window, right click **Inbound Rules** and click **New Rule**.
4. In the **New Inbound Rule Wizard** dialog box, Select the type of rule you want to create, and then Click **Next** and browse to the location of msdtc.exe in the .
5. The next step **Program** appears, Browse to the location of msdtc.exe in **This program path** to apply the rule to a specific program.
6. Provide appropriate values to the steps in the **New Inbound Rule Wizard** window, and then click **OK**.

See Also

Other Resources

[Before Developing Applications Using the Oracle E-Business Suite Adapter](#)

Developing BizTalk Applications

Developing BizTalk applications involves creating a BizTalk project in Visual Studio and using the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard to generate XML schema. Once you have generated the schema, you can either use Content-Based Routing (CBR) or create BizTalk orchestrations to send and receive messages that conform to the generated schema.

CBR can be used in scenarios where the messages being sent to Oracle E-Business Suite do not require any intensive processing. For example, if you know that the receive port will receive messages only of a certain type, you can add a filter to the send port to route messages that match the filter expression to the send port.

In BizTalk orchestrations, you create send and receive ports to send and receive messages to and from the WCF-Custom adapter, which in turn sends the messages to BizTalk Server. This section provides information about using BizTalk orchestrations to perform operations on Oracle E-Business Suite using the Oracle E-Business adapter. The Oracle E-Business adapter in turn uses the WCF-Custom adapter, which can interact with a WCF binding.

◆ Important

To use the Oracle E-Business adapter with Microsoft BizTalk Server, you must always set the **EnableBizTalkCompatibility Mode** binding property to **True**. For information about how to set binding properties, see [Specifying Binding Properties](#).

In This Section

- [Before You Develop BizTalk Applications](#)
- [Building Blocks for Developing BizTalk Applications](#)
- [Performing Insert, Update, Delete, or Select Operations on Interface Tables and Interface Views Using BizTalk Server](#)
- [Performing Operations on Tables with Large Data Types Using BizTalk Server](#)
- [Performing Composite Operations on Oracle Database by Using BizTalk Server](#)
- [Invoking Concurrent Programs in Oracle E-Business Suite Using BizTalk Server](#)
- [Invoking Request Sets in Oracle E-Business Suite Using BizTalk Server](#)
- [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server](#)
- [Polling Oracle E-Business Suite Using BizTalk Server](#)
- [Receiving Database Change Notifications Using BizTalk Server](#)

See Also

Other Resources
[Development](#)

Before You Develop BizTalk Applications

This section provides information about what you must do before developing BizTalk applications using the Oracle E-Business adapter. This section also points you to the BizTalk Server tools that are used to develop BizTalk applications.

In This Section

- [Create a Strong-Name Key File](#)
- [Learn About BizTalk Server Tools](#)

Create a Strong-Name Key File

You must create a strong-name key file to build projects in Microsoft Visual Studio. A strong name consists of the project's identity—its simple text name, version number, and culture information (if provided)—plus a public key and a digital signature. The strong-name key file contains the public key and the private key.

◆ Important

Creating a strong-name key file is a one-time task. You can use the same key for all the BizTalk applications you develop.

Prerequisites

You must have Microsoft Visual Studio installed on the computer where you want to create a strong-name key.

To create a strong-name key file

1. Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2010**, point to **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
2. At the command prompt, navigate to the location where you want to create the key file. For example, type **cd C:\Sample**, and then press ENTER.
3. At the command prompt, type **sn -k <key file name>.snk**, and then press ENTER.

📌 Note

You should receive a message at the command prompt stating that the key pair was written to the strong-name key file.

4. At the command prompt, type **exit**, and then press ENTER.

See Also

Concepts

[Learn About BizTalk Server Tools](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Learn About BizTalk Server Tools

The topics on how to use the Oracle E-Business adapter in [Developing BizTalk Applications](#) are written with the assumption that you have working knowledge of a number of BizTalk Server tools. You will use the following tools to develop BizTalk applications using the Oracle E-Business adapter:

- Microsoft Visual Studio 2010
- BizTalk Explorer
- Orchestration Designer
- Pipeline Designer
- BizTalk Mapper
- BizTalk Server Administration console

Prerequisites

You must install Microsoft BizTalk Server before you can access the BizTalk Server tools.

BizTalk Server Tools

The following table includes topics in the BizTalk Server documentation that explain how to use each of the listed tools.

Tool	Topics in BizTalk Server Documentation
Visual Studio	<ul style="list-style-type: none"> • Using Visual Studio (http://go.microsoft.com/fwlink/?LinkId=107180) • Working with BizTalk Projects (http://go.microsoft.com/fwlink/?LinkId=107181) • Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=107182) <p>Learn more about Visual Studio:</p> <ul style="list-style-type: none"> • Introduction to Solutions, Projects, and Items (http://go.microsoft.com/fwlink/?LinkID=196962) • Solution Explorer Concepts (http://go.microsoft.com/fwlink/?LinkID=196962)
Orchestration Designer	<ul style="list-style-type: none"> • Creating Orchestrations Using Orchestration Designer (http://go.microsoft.com/fwlink/?LinkId=107184)
Pipeline Designer	<ul style="list-style-type: none"> • Creating Pipelines Using Pipeline Designer (http://go.microsoft.com/fwlink/?LinkId=107185)
BizTalk Mapper	<ul style="list-style-type: none"> • Creating Maps Using BizTalk Mapper (http://go.microsoft.com/fwlink/?LinkId=107186)
BizTalk Server Administration console	<ul style="list-style-type: none"> • Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkId=107187)

See Also

Tasks

[Create a Strong-Name Key File](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Building Blocks for Developing BizTalk Applications

To perform operations on Oracle E-Business Suite by using the Oracle E-Business adapter with BizTalk Server, you must perform a set of design-time and run-time tasks using Visual Studio and BizTalk Server Administration console respectively. This section provides an overview of these tasks. All the topics in this section, which demonstrate how to perform specific operations on Oracle E-Business Suite using BizTalk Server, are modeled on these high-level tasks.

Using Visual Studio

1. **Create BizTalk project, and generate schema.** You must create a BizTalk project in Visual Studio, and generate schema for the operation that you will perform on Oracle E-Business Suite. For example, if you want to select records from an Oracle E-Business Suite interface table, you must generate schema for the Select operation for that table. To generate schema, you must use the Consume Adapter Service Add-in or Add Adapter Metadata Wizard. For more information, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).
2. **Set up an orchestration.** Once you have generated the schema, you must set up an orchestration by using the Orchestration Designer. For a basic orchestration, you add the Send and Receive shapes along with the Send and Receive logical ports. In later steps, you map these logical ports to physical ports by using the BizTalk Server Administration console. The orchestration uses these ports to pick messages that an adapter client sends. The orchestration then passes the messages to Oracle E-Business Suite. Once Oracle E-Business Suite sends a response, the orchestration passes the response back to the adapter client.
3. **Create messages, and link to schema.** In your orchestration, you must create messages that will be mapped to the schema you generated in the first step. Typically, you create a request message and a response message. These messages are mapped to the corresponding request and response schemas.
4. **Map message shapes to messages and ports.** In your orchestration, you must now map each shape that you added in the second step to messages that you created in the third step. You must also map a message shape to the port on which that message will be sent.

For example, if the first shape in your orchestration is a Receive shape that will receive a message, you map this shape to a request message and the port that sends the request message.
5. **Build and deploy the BizTalk project.** After you have set up the orchestration and mapped messages, ports, and schemas, you must build the BizTalk solution. For building a project in Visual Studio, you need an assembly key file. After you successfully build the solution, you must deploy the solution.

Note

More detailed description of these high-level tasks, including procedural information, is provided in various topics of this section.

Once you have successfully built and deployed the BizTalk project, your design-time tasks in Visual Studio are accomplished. You must now perform certain run-time tasks using the BizTalk Server Administration console.

Using the BizTalk Server Administration Console

1. **Configure the application.** The BizTalk project you deployed by using Visual Studio shows up in the BizTalk Server Administration console as an orchestration. You must configure this orchestration by mapping the logical ports you created in Visual Studio to physical ports that you must now create using the BizTalk Server Administration console.

On the physical ports, you must specify an "action" or "action mapping". This action corresponds to the operation you want to perform on Oracle E-Business Suite. You need to specify the action if you are not using dynamic actions. For more information about actions, see [Specifying SOAP Action](#).

2. **Start the application.** After the application is configured, you must start the application, and drop request messages at a defined file location. The orchestration consumes the request messages, passes them to Oracle E-Business Suite, and

receives a response. This response is available to the adapter client at another defined file location.

To accomplish these high-level tasks, you must also perform other tasks. For example, when you use the Consume Adapter Service Add-in or Add Adapter Metadata Wizard to generate the schema, you must specify a connection URI to connect to Oracle E-Business Suite. This section provides information on such repetitive tasks that you must perform as you develop BizTalk applications using the Oracle E-Business adapter.

In This Section

- [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#)
- [Specifying the Connection URI](#)
- [Specifying Credentials for the Oracle E-Business Suite](#)
- [Specifying Binding Properties](#)
- [Specifying SOAP Action](#)
- [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#)
- [Configuring a Physical Port Binding Using a Port Binding File](#)
- [Configuring Dynamic Ports](#)
- [Reusing Adapter Bindings](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console

The Oracle E-Business adapter can be used in BizTalk Server either as a WCF-Custom port or a WCF-OracleEBS port. If you want to use the Oracle E-Business adapter through a WCF-Custom port, you do not need to add the WCF-Custom port to the BizTalk Server Administration console because the WCF-Custom port is added to the BizTalk Server Administration console by default. However, if you want to use the Oracle E-Business adapter through a WCF-OracleEBS port, you must first add the WCF-OracleEBS adapter to the BizTalk Server Administration console.

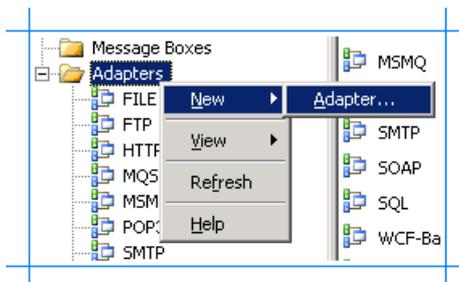
This topic provides instructions on how to add the WCF-OracleEBS adapter to the BizTalk Server Administration console.

Important

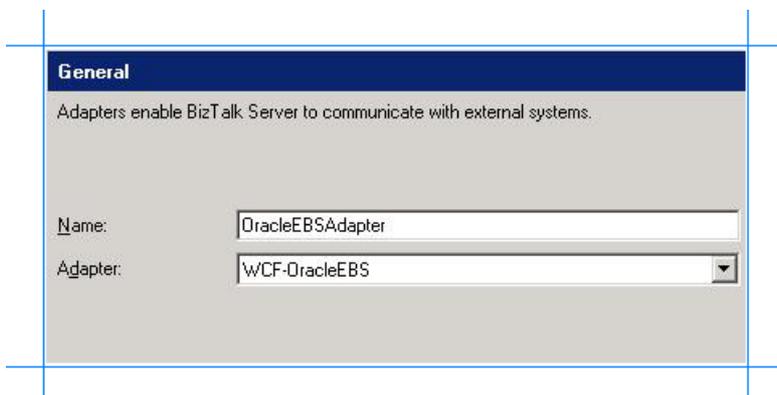
If you want to configure a WCF-Custom port for the Oracle E-Business adapter, you need not perform this procedure.

To add the Oracle E-Business Suite Adapter

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, expand **Platform Settings**, and then click **Adapters**.
3. Right-click **Adapters**, point to **New**, and click **Adapter**.



4. In the **Adapter Properties** dialog box, specify a name for the adapter and from the **Adapter** list, select **WCF-OracleEBS**.



5. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying the Connection URI

A connection URI is a connection string that contains parameters required to connect to Oracle E-Business Suite. While using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard in Visual Studio, you must specify the URI to connect to Oracle E-Business Suite to generate the metadata. While configuring an orchestration using the BizTalk Server Administration console, you must specify the URI to connect to Oracle E-Business Suite to perform operations.

Specifying the Connection URI from Visual Studio

From Visual Studio, you must specify the connection URI using the Consume Adapter Service Add-in or Add Adapter Metadata Wizard.

To specify the connection URI using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleEBSBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab. From the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

6. Click the **URI Properties** tab, and specify values for different parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).
7. Click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
8. Click **OK**.

To specify the connection URI using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleEBS**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-OracleEBS port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleEBSBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab. From the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

8. Click the **URI Properties** tab, and specify values for different parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).
9. Click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

📌 Note
If you selected an existing WCF-OracleEBS send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

10. Click **OK**.

Specifying the Connection URI from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the connection URI as part of the WCF-Custom or WCF-OracleEBS port configuration.

To specify the connection URI for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Address (URI)** text box, specify the connection URI to connect to Oracle E-Business Suite. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).
6. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **oracleEBSBinding**.
7. If you are creating a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
8. If you are creating a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab, and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

9. Click **OK**.

To specify the connection URI for the WCF-OracleEBS port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-OracleEBS adapter you add earlier, and then click **Configure**.

 **Note**

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the transport properties dialog box, click the **General** tab.
6. Click the **Configure** button and provide values for the connection parameters. For more information about the connection URI for the Oracle E-Business adapter, see [The Oracle E-Business Suite Connection URI](#).
7. In the transport properties dialog box, click the **Binding** tab and specify values for binding properties.

 **Note**

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

8. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:

- Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

9. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

10. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Specifying Credentials for the Oracle E-Business Suite

The Oracle E-Business adapter requires the adapter clients to provide client credentials. The adapter uses these credentials to authenticate the user with the Oracle E-Business Suite and to establish a connection.

Adapter clients can provide the client credentials both when using Visual Studio and when using the BizTalk Server Administration console. When using Visual Studio, credentials are required to generate the metadata. When using the BizTalk Server Administration console, credentials are required to perform operations on Oracle E-Business Suite.

◆ Important

You can specify the credentials for the Oracle E-Business Suite or the underlying Oracle database. To connect and generate metadata you can specify any credentials. However, while performing an operation to invoke an Oracle E-Business Suite artifact, you must specify the Oracle E-Business Suite credentials because they are required to set the application context for the Oracle E-Business Suite application you want to invoke. For more information about setting applications context, see [Setting Application Context](#).

This section provides information about specifying client credentials in Visual Studio and the BizTalk Server Administration console.

Specifying Credentials from Visual Studio

From Visual Studio, you must specify the credentials using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify credentials using Consume Adapter Service Add-in

1. Right-click your BizTalk project, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleEBSBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.

To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.
--	---

6. Click **OK**.

To specify credentials using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleEBS**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-OracleEBS port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleEBSBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, select **Username** and specify the user name and password to connect to the Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

8. Click **OK**.

Specifying Credentials from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the credentials as part of the WCF-Custom or WCF-OracleEBS port configuration.

To specify credentials for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note
To see the location properties dialog box for a receive port, click the Receive Location tab on the left pane of the port properties dialog box, and then click New .

4. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **oracleEBSBinding**.
5. If you are creating a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

6. If you are creating a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.

To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

7. Click **OK**.

To specify credentials for the WCF-OracleEBS port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select **WCF-OracleEBS**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the port properties dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **oracleEBSBinding**.
6. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

7. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if Client CredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if Client CredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

8. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Specifying Binding Properties

The Oracle E-Business adapter surfaces several binding properties that enable you to control some of its behavioral characteristics. This section provides information about setting the binding properties from Visual Studio and from the BizTalk Server Administration console. From Visual Studio, you must specify the binding properties while generating schema for specific operations. From BizTalk Server, you must specify the binding properties as part of the send or receive port for sending or receiving messages from Oracle E-Business Suite.

For information about the binding properties, including a list of binding properties for the Oracle E-Business adapter, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Specifying Binding Properties from Visual Studio

From Visual Studio, you must specify the binding properties using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify binding properties using Consume Adapter Service Add-in

1. Right-click your BizTalk project, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list, select **oracleEBSBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Binding Properties** tab, and then specify the different binding properties.
6. Click **OK**.

To specify binding properties using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Metadata Wizard, select **WCF-OracleEBS**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important

If you already have a WCF-OracleEBS port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **oracleEBSBinding**, and then click **Configure**.
7. Click the **Binding Properties** tab, and specify the binding values, if any, which are required before generating the

schema. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Note

If you selected an existing WCF-OracleEBS send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such a case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

8. Click **OK**.

Specifying Binding Properties from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the binding properties as part of the WCF-Custom or WCF-OracleEBS port configuration.

To specify binding properties for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

4. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab.
5. From the **Binding Type** list, select **oracleEBSBinding**.
6. In the **Configuration** box, specify the values for the different binding properties, and then click **OK**.

To specify binding properties for the WCF-OracleEBS port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-OracleEBS adapter you added earlier, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the transport properties dialog box, click the **Binding** tab and specify values for binding properties.

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying SOAP Action

To perform any operation on Oracle E-Business Suite using the WCF-based Oracle E-Business adapter, you must specify a SOAP action. The SOAP action communicates to the adapter what action should be performed. You can specify the SOAP action either from Visual Studio or from the BizTalk Server Administration console. However, if you specify the SOAP action from both locations, the action you specified from Visual Studio will be overridden.

For more information about specifying SOAP action, see <http://go.microsoft.com/fwlink/?LinkId=102063>.

Specifying SOAP Action from Visual Studio

From Visual Studio, you must specify the SOAP action as part of the orchestration by using an **Expression** shape.

To specify SOAP action from Visual Studio

1. In the BizTalk orchestration, include an **Expression** shape by dragging it from the **BizTalk Orchestration** toolbox.
2. Double-click the **Expression** shape to open BizTalk Expression Editor.
3. Specify the action in BizTalk Expression Editor. For example:

```
OutboundMessage(WCF.Action)="InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY"
```

For more information about the **Expression** shape and BizTalk Expression Editor, see <http://go.microsoft.com/fwlink/?LinkId=102064>.

Specifying SOAP Action from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you must specify the SOAP action as part of the WCF-Custom or WCF-OracleEBS port configuration.

To specify a SOAP action for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:

- **By using the single action format.** Use this format if the WCF-Custom port sends and receive messages for a single operation. For example:

```
InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY
```

- **By using the action mapping format.** Use this format if a single WCF-Custom port sends and receives messages for more than one operation. For example, if a single WCF-Custom port sends and receives messages for Op1 (to insert records in the GL_ALLOC_HISTORY table) and Op2 (to update records in the GL_ALLOC_HISTORY table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY"
  />
  <Operation Name="Op2" Action="InterfaceTables/Update/SQLGL/GL/GL_ALLOC_HISTORY "
  />
</BtsActionMapping>
```

The action mapping approach provides greater flexibility in terms of specifying a set of actions, and hence enabling

messages that belong to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about the action format for each operation, see individual topics under [Technical Reference](#).

To specify a SOAP action for the WCF-OracleEBS port

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-OracleEBS adapter you add earlier, and then click **Configure**.
5. In the transport properties dialog box, click the **General** tab.
6. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:
 - **By using the single action format.** Use this format if the WCF-OracleEBS port sends and receive messages for a single operation. For example:

```
InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY
```

- **By using the action mapping format.** Use this format if a single WCF-OracleEBS port sends and receives messages for more than one operation. For example, if a single WCF-OracleEBS port sends and receives messages for Op1 (to insert records in the GL_ALLOC_HISTORY table) and Op2 (to update records in the GL_ALLOC_HISTORY table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>  
  <Operation Name="Op1" Action="InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY"  
  />  
  <Operation Name="Op2" Action="InterfaceTables/Update/SQLGL/GL/GL_ALLOC_HISTORY "  
  />  
</BtsActionMapping>
```

The action mapping approach provides greater flexibility in terms of specifying a set of actions, and hence enabling messages that belong to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about the action format for each operation, see individual topics under [Technical Reference](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter

This section provides information about configuring the Microsoft BizTalk Adapter for Oracle E-Business Suite as a WCF custom binding or as a WCF-OracleEBS port by using the BizTalk Server Administration console. After deploying the adapter, you will be able to send and receive messages from Oracle E-Business Suite by using the BizTalk Server Administration console. The steps for deploying the adapter vary depending on:

- The direction of communication between BizTalk Server and the Oracle E-Business adapter. You may choose to configure a send, receive, send-receive, or a receive-send port. Your choices are summarized in the following table.

Port direction	Communication pattern	Direction of communication to choose from
Send	One-way	I will always be sending messages on this port.
Receive	One-way	I will always be receiving messages on this port.
Send-receive	Request-response	I will be sending a request and receiving a response.
Receive-send	Solicit-response	I will be receiving a request and sending a response.

For more information, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

- Whether the adapter sends messages to or receives messages from Oracle E-Business Suite. Depending on whether you want to send or receive messages, you will create a send or receive port, respectively.

Note

You can also configure the send or receive ports by importing a binding configuration file that is created by the Consumer Adapter Service Add-in as part of metadata generation. For instructions on configuring ports using this binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

In This Section

- [Configuring a Port Using the WCF-Custom Adapter](#)
- [Configuring a Port Using the WCF-OracleEBS Adapter](#)

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring a Port Using the WCF-Custom Adapter

This topic provides instructions on how to configure WCF-Custom send and receive ports to perform outbound and inbound operations on Oracle E-Business Suite using the Oracle E-Business adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to Oracle E-Business Suite

Perform the following steps to configure a WCF-Custom send port for sending messages to Oracle E-Business Suite using the BizTalk Server Administration console.

To deploy adapters for sending messages to Oracle E-Business Suite

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the Oracle E-Business adapter.
4. Right-click **Send Ports**, point to **New**, and then point to the type of port you want to configure depending on the mode of communication between BizTalk Server and Oracle E-Business Suite.
5. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
6. From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
7. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for Oracle E-Business Suite. For more information about the connection URI, see [The Oracle E-Business Suite Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on an interface table (FA_BOOKS) under the Asset application is:

```
InterfaceTables/Insert/OFA/FA/FA_BOOKS
```

- c. Click the **Binding** tab, and from the **Binding Type** list, select **oracleEBSBinding**. You can specify the different binding properties exposed by the Oracle E-Business adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).
- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.

To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

e. To return to the **Send Port Properties** dialog box, click **OK**.

- From the **Send handler** list, select **BizTalkServerApplication**.
- If you chose **Static One-Way Send Port** in step 4, specify a send pipeline. From the **Send pipeline** list, select the pipeline that corresponds to XMLTransmit.
- If you chose **Static Solicit-Response Port** in step 4, specify send and receive pipelines.
 - From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
- Click **OK**.

Deploying Adapters for Receiving Messages from Oracle E-Business Suite

Perform the following steps to configure a WCF-Custom receive port for receiving messages from Oracle E-Business Suite using the BizTalk Server Administration console.

To deploy adapters for receiving messages from Oracle E-Business Suite

- Start the BizTalk Server Administration console.
- In the console tree, expand **BizTalk Group**, and then expand **Applications**.
- Expand the application under which you want to deploy the Oracle E-Business adapter.
- Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and Oracle E-Business Suite.
- In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
- On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
- In the **Receive Location Properties** dialog box, do the following:
 - Specify a name for the receive location.
 - From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
- In the **WCF-Custom Transport Properties** dialog box, do the following:
 - Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for Oracle E-Business Suite. For more information about the connection URI, see [The Oracle E-Business Suite Connection URI](#).
 - Click the **Binding** tab, and from the **Binding Type** drop-down list, select **oracleEBSBinding**. You can specify the different binding properties exposed by the Oracle E-Business adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

c. Click the **Behavior** tab to set the transaction isolation level. For more information about setting transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).

d. Click the **Others** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

e. To return to the **Receive Location Properties** dialog box, click **OK**.

9. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.
10. If you chose **One-way Receive Port** in step 4, specify a receive pipeline. From the **Receive pipeline** list, select the pipeline corresponding to XMLReceive.
11. If you chose **Request Response Receive Port** in step 4, specify send and receive pipelines.
 - a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
12. In the **Receive Location Properties** dialog box, click **OK**.
13. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Other Resources

[Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#)

Configuring a Port Using the WCF-OracleEBS Adapter

This topic provides instructions on how to configure WCF-OracleEBS send and receive ports to perform outbound and inbound operations on Oracle E-Business Suite using the Oracle E-Business adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to Oracle E-Business Suite

Perform the following steps to configure a WCF-OracleEBS send port for sending messages to Oracle E-Business Suite using the BizTalk Server Administration console.

To deploy adapters for sending messages to Oracle E-Business Suite

1. Start the BizTalk Server Administration console.
2. Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
4. Expand the application under which you want to deploy the Oracle E-Business adapter.
5. Right-click **Send Ports**, point to **New**, and then point to the type of port you want to configure depending on the mode of communication between BizTalk Server and Oracle E-Business Suite.
6. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
7. From the **Type** drop-down list, select WCF-OracleEBS, and then click **Configure**.
8. In the transport properties dialog box, do the following:
 - a. Click the **General** tab, click the **Configure** button and provide values for the connection parameters. For more information about the connection URI, see [The Oracle E-Business Suite Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on an interface table (FA_BOOKS) under the Asset application is:

```
InterfaceTables/Insert/OFA/FA/FA_BOOKS
```

- c. Click the **Binding** tab and specify values for the binding properties exposed by the Oracle E-Business adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
----------	------------

To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

e. To return to the **Send Port Properties** dialog box, click **OK**.

- From the **Send handler** list, select **BizTalkServerApplication**.
- If you chose **Static One-Way Send Port** in step 5, specify a send pipeline. From the **Send pipeline** list, select the pipeline that corresponds to XMLTransmit.
- If you chose **Static Solicit-Response Port** in step 4, specify send and receive pipelines.
 - From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
- Click **OK**.

Deploying Adapters for Receiving Messages from Oracle E-Business Suite

Perform the following steps to configure a WCF-OracleEBS receive port for receiving messages from Oracle E-Business Suite using the BizTalk Server Administration console.

To deploy adapters for receiving messages from Oracle E-Business Suite

- Start the BizTalk Server Administration console.
- Add the WCF-OracleEBS adapter to the BizTalk Server Administration console. For instructions, see [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).
- In the console tree, expand **BizTalk Group**, and then expand **Applications**.
- Expand the application under which you want to deploy the Oracle E-Business adapter.
- Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and Oracle E-Business Suite.
- In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
- On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
- In the **Receive Location Properties** dialog box, do the following:
 - Specify a name for the receive location.
 - From the **Type** drop-down list, select WCF-OracleEBS, and then click **Configure**.

9. In the transport properties dialog box, do the following:

- a. Click the **General** tab, click the **Configure** button, and provide values for the connection parameters. For more information about the connection URI, see [The Oracle E-Business Suite Connection URI](#).
- b. Click the **Binding** tab and specify values for binding properties exposed by the Oracle E-Business adapter. For more information about binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- c. Click the **Behavior** tab to set the transaction isolation level. For more information about setting transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).
- d. Click the **Others** tab, and do one of the following:

- Select **User account** option, and specify the user name and password to connect to Oracle E-Business Suite.

Use this	To do this
To connect using Oracle database credentials	Specify the ClientCredentialType binding property to Database and specify database credentials for User name and Password text boxes.
To connect using Oracle E-Business Suite credentials	Specify the ClientCredentialType binding property to EBusiness and specify Oracle E-Business Suite credentials for User name and Password text boxes. In this case, you must also specify Oracle database credentials for OracleUserName and OraclePassword binding properties.
To connect using Windows Authentication if ClientCredentialType is set to "Database"	Specify a "/" for the User name text box and leave the Password text box blank.
To connect using Windows Authentication if ClientCredentialType is set to "EBusiness"	Specify Oracle E-Business Suite credentials for User name and Password text boxes. You must also specify a "/" for the OracleUserName binding property and leave the OraclePassword binding property blank.

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

e. To return to the **Receive Location Properties** dialog box, click **OK**.

10. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.

11. If you chose **One-way Receive Port** in step 5, specify a receive pipeline. From the **Receive pipeline** list, select the pipeline corresponding to XMLReceive.

12. If you chose **Request Response Receive Port** in step 5, specify send and receive pipelines.

- a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
- b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.

13. In the **Receive Location Properties** dialog box, click **OK**.

14. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Concepts

[Connecting to Oracle E-Business Suite Using Windows Authentication](#)

Other Resources

[Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#)

Configuring a Physical Port Binding Using a Port Binding File

When you use the Consume Adapter Service BizTalk Project Add-in to generate metadata for Oracle E-Business Suite artifacts, other than the schema files, the Consume Adapter Service Add-in also generates a port binding file. You can import this binding file into your BizTalk application to create a physical send or receive port. For instructions on importing binding files, see [Importing Bindings](#). If you import this binding file, you do not have to manually create a physical send or receive port.

Important

While using the Consume Adapter Service BizTalk Project Add-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file. You must manually add the binding property and its value in the binding file, if required.

Creating a port using the port binding file always creates a two-way send port or a one-way receive port. If you want to create a one-way send port, you can create it manually by following the procedure mentioned in [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). Or you can follow the workarounds documented in this topic to modify the port binding file to create one-way send ports.

Note

For inbound operations, the port binding file will always create a one-way receive port. This is because the Oracle E-Business adapter only supports one-way receive port for inbound operations.

Important

Using the Add Adapter Metadata Wizard does not create a port binding file using which you can create a WCF-OracleEBS port. However, you could make some changes to the port binding file generated by the Consume Adapter Service Add-in and use it to create a WCF-OracleEBS port. For more information, see [Configuring a WCF-OracleEBS Port Using the Port Binding File Generated Using Consume Adapter Service Add-in](#).

Following are some key points that you must understand with respect to the binding file that the Consume Adapter Service Add-in generates:

- The files are created with a specific naming convention. If you generated metadata for outbound operations, that is, to send messages to Oracle E-Business Suite, the name of the file is WcfSendPort_OracleEBSBinding_Custom.bindinginfo.xml.

If you generated metadata for inbound operations, that is, to receive messages from Oracle E-Business Suite, the name of the file is WcfReceivePort_OracleEBSBinding_Custom.bindinginfo.xml.

- The file contains information about the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. When you import this binding file into your BizTalk application to create a port, all the relevant information required to configure a physical port is automatically set on the port.

Important

By default, the action on the send port is mapped to the operation name for which you generated metadata. For example, if you generate metadata for the Insert operation on the interface table (e.g. FA_BOOKS), the action on the port is set to `<Operation Name="Insert" Action="InterfaceTables/Insert/OFA/FA/FA_BOOKS" />`. However, the operation name on the logical send port you create in the BizTalk orchestration might not be the same. You must ensure that the operation name in the logical send port (in the BizTalk orchestration) and the physical send port (in BizTalk Server Administration console) are the same. If not, you will receive an error while sending messages to Oracle E-Business Suite through the send port.

- You only need to provide the credentials for the port to connect to Oracle E-Business Suite. Although the binding file does retain the user name with which to connect, for security reasons the binding file does not contain the password.

Key Considerations for Using the Port Binding File

- When you import the binding file, you might get a dialog message informing that the BizTalk application name in the binding file does not match the application name to which you are importing the binding file. You can safely ignore this message and continue.
- The binding file also contains names of ports and receive locations. If the BizTalk application to which you are importing the binding file creates a port or a receive location that has the same name as an already existing port in the same BizTalk application, you will get an error. You must manually edit the binding file to specify a unique name for the ports or receive locations.
- The binding file also contains information about the connection URI. If the binding file creates a receive location that has the same receive URI as an already existing receive location in the same BizTalk application, you will get an error. You must manually edit the binding file to specify a unique URI.
- By default, the port binding file for outbound operations always contains definitions for a two-way send port. When you import this file in a BizTalk application, it creates a two-way send port. However, you may have an orchestration that has a one-way send port. So, when you configure such an orchestration and use the port created by importing the binding file, the port is not available in the list. This happens because the logical port you created as part of the orchestration is a one-way port while the physical port created in the orchestration is a two-way port. In such cases, you can edit the binding file to make the following changes:

For this	Do this
<p>To edit the port binding file to configure a one-way send port</p>	<ol style="list-style-type: none"> 1. In the following excerpt, change the value of the IsTwoWay property to false. Originally, this is set to true. <div data-bbox="687 1019 1497 1137" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><SendPort Name="port_name" IsStatic="true" IsTwoWay="false" BindingOption="0"></pre> </div> 2. Comment out the following excerpts: <div data-bbox="687 1227 1497 1563" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipeline Name="Microsoft.BizTalk.DefaultPipelines.XMLReceive" FullyQualifiedName="Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=token" Type="1" TrackingOption="None" Description="" /></pre> </div> <div data-bbox="687 1585 1497 1664" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipelineData xsi:nil="true" /></pre> </div>

◆ Important

For inbound operations, the port binding file will always create a one-way receive port. This is because the Oracle E-Business adapter only supports one-way receive port for inbound operations.

Configuring a WCF-OracleEBS Port Using the Port Binding File Generated Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in creates a port binding file that you can import in BizTalk Server Administration console. You can use the same port binding file to also create the BizTalk WCF-OracleEBS port in BizTalk Server Administration console. However, before creating WCF-OracleEBS port you must perform the following tasks to modify the port binding file.

1. Open the port binding file in a text editor.
2. Search and replace "WCF-Custom" with the name with which you added the WCF-OracleEBS adapter in BizTalk Server Administration console. For example, if you added the WCF-OracleEBS adapter as "OracleEBSAdapter", replace "WCF-Custom" with "OracleEBSAdapter".
3. Search for the "ConfigurationClsid" attribute, and replace the existing value of the attribute with "F452BB15-7A0D-495d-9395-C630D3FD29CD".
4. Save and close the binding file.
5. Import the binding file in BizTalk Server Administration console. For instructions on how to import the binding file, see [Importing Bindings](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring Dynamic Ports

In BizTalk Server, you can configure dynamic ports for a WCF-Custom adapter. Because the Oracle E-Business adapter is a WCF-based adapter, you can dynamically configure a port for the Oracle E-Business adapter by using message context properties.

For the Oracle E-Business adapter, the URI, action, and binding may be determined from a property on an incoming message, and then specified in the **Expression** shape, as shown in the following example:

```
Request2=Request1;  
Request2(WCF.Action)="InterfaceTables/Insert/OFA/FA/FA_BOOKS";  
Request2(WCF.BindingType)="oracleEBSBinding";  
Request2(WCF.UserName)="myuser";  
Request2(WCF.Password)="mypass";  
SendPort(Microsoft.XLANGs.BaseTypes.Address)="oracleebs://ebs_instance";  
SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="WCF-Custom";
```

Note

If you are using a WCF-OracleEBS adapter in BizTalk Server Administration console, you can also specify the transport type as `SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="OracleEBSAdapter"`, where **OracleEBSAdapter** is the name with which you added the WCF-OracleEBS adapter in BizTalk Server Administration console.

In the preceding example,

- Request2 message is being created from Request1 message. Both messages map to an operation schema, which is generated using the Consume Adapter Service Add-in.
- SendPort is the name of the logical send port in the BizTalk orchestration.

The **Expression** shape is part of the BizTalk orchestration. Deploying the orchestration also creates a WCF-Custom send port.

For more information about configuring dynamic ports, see "Configuring Dynamic Send Ports Using WCF Adapters Context Properties" at <http://go.microsoft.com/fwlink/?LinkId=106735>.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Reusing Adapter Bindings

A binding creates a mapping between a logical endpoint (such as an orchestration port or a role link) and a physical endpoint (such as a send and receive port). This enables communication between different components of a BizTalk Server business solution. You can create bindings by using the BizTalk Server Administration console.

What is a Binding File?

A binding file is an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. The binding file describes:

- The host to which each orchestration is bound.
- The trust level of the host.
- The settings for each send port, receive port, receive location, and party that has been configured.

You can generate binding files and then apply the bindings that they contain to an assembly, application, or group. This prevents having to manually configure bindings in different deployment environments and speeds up application deployment.

A binding file is not automatically generated for a BizTalk assembly, application, or group. However, you can generate a binding file by exporting bindings. Similarly, you can then import the binding file into an application or group. This section provides instructions on how to import and export bindings.

For more information about bindings and about binding files, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

In This Section

- [Exporting Bindings](#)
- [Importing Bindings](#)

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Exporting Bindings

This section describes how to export bindings for a BizTalk application into an XML file. You can then import the bindings from the XML file into another BizTalk application. Importing bindings overwrites any existing bindings of the same name in the application. You can also add bindings to an application, which does not overwrite existing bindings. The bindings that you add do not take effect until you import the application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To export bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application whose bindings you want to export, point to **Export**, and then click **Bindings**.
4. On the **Export Bindings** page, in **Export to file**, type the absolute path of the XML file to which to export the bindings.
Example: C:\Bindings\Application1Bindings.xml.
5. Ensure that **Export all bindings from the current application** is selected.
6. To export all party information for the group, select the **Export Global Party information** check box.
7. Click **OK**.

The bindings are exported into an XML file in the location that you specified.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the **Transport Properties** dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for the Oracle E-Business Suite](#).

See Also

Concepts

[Reusing Adapter Bindings](#)

Importing Bindings

This topic describes how to use the BizTalk Server Administration console to import bindings into a BizTalk application from an XML file.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To import bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application into which you want to import bindings, point to **Import**, and then click **Bindings**.
4. Click the binding file, and then click **Open**.

The artifacts in the binding file are written to the application. They are displayed in appropriate folders of the application. For example, the send ports that are imported as part of the bindings are displayed under the Send Ports folder.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the **Transport Properties** dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for the Oracle E-Business Suite](#).

See Also

Concepts

[Reusing Adapter Bindings](#)

Performing Insert, Update, Delete, or Select Operations on Interface Tables and Interface Views Using BizTalk Server

The Oracle E-Business adapter surfaces a set of standard operations such as Insert, Update, Delete, Select on interface tables and views. This topic provides instructions on how to perform these operations using the adapter. For more information about how the adapter supports these operations, see [Operations on Interface Tables and Interface Views](#). For information about the structure of the SOAP message for these operations, see [Message Schemas for Insert, Update, Delete, and Select Operations](#).

Note

The adapter also exposes specific operations for tables and views that contain large data types such as BLOB, CLOB, NCLOB, and BFILE. For more information about such operations, see [Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#). For instructions on how to perform operations on tables and columns with large data types using BizTalk Server, see [Performing Operations on Tables with Large Data Types Using BizTalk Server](#).

How to Perform Basic Operations on Oracle E-Business Suite

Performing an operation on Oracle E-Business Suite by using Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform Insert, Update, Delete, or Select operations on tables and views in Oracle E-Business Suite, these tasks are:

1. Create a BizTalk project, and generate schema for the operation you want to invoke on an interface table or view.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle E-Business Suite.
3. Create an orchestration to invoke the operation on the interface table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to perform basic Insert, Update, Delete, or Select operations by inserting records in the AR_ARCHIVE_PURGE_INTERIM interface table in Oracle E-Business Suite. This interface table is available in the **Receivables** application in the Oracle E-Business Suite.

To demonstrate how to insert records, schema is generated for the Insert operation for the AR_ARCHIVE_PURGE_INTERIM table. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) for more information about how to generate schemas.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.

- In Orchestration View, right-click **Messages**, and then click **New Message**.
- Right-click the newly created message, and then select **Properties Window**.
- In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>InsertInterfaceTable.OracleEBSBinding.Insert</i> , where <i>InsertInterfaceTable</i> is the name of your BizTalk project. <i>OracleEBSBinding</i> is the schema generated for the Insert operation on the AR_ARCHIVE_PURGE_INTERIM table.

- Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

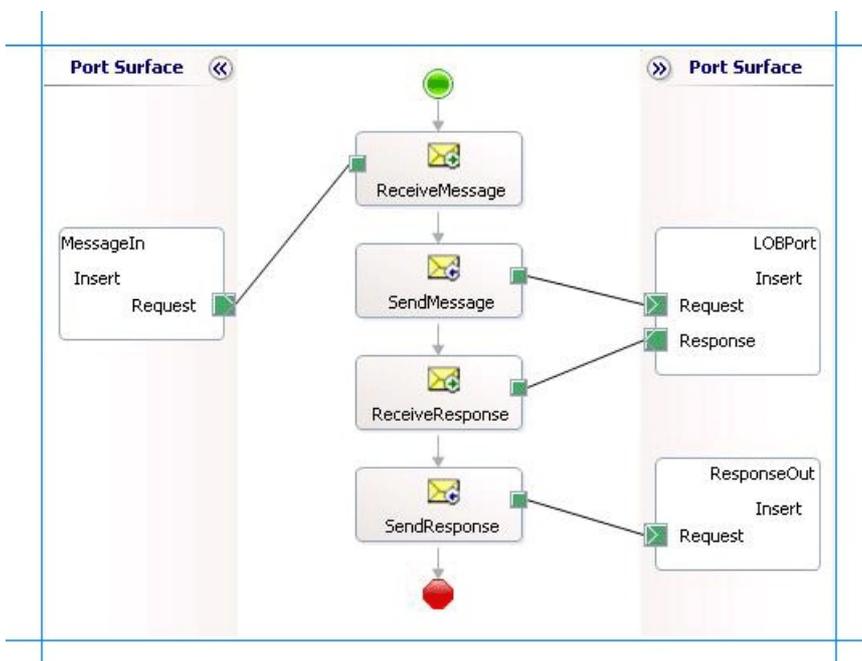
Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>InsertInterfaceTable.OracleEBSBinding.InsertResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on Oracle E-Business Suite. In this orchestration, you drop a request message at a defined receive location. The Oracle E-Business adapter consumes this message and passes it on to Oracle E-Business Suite. The response from Oracle E-Business Suite is saved to another location. A typical orchestration for performing basic table operations on Oracle database would contain:

- Send and Receive shapes to send messages to Oracle database and receive responses.
- A one-way receive port to receive request messages to send to the Oracle database.
- A two-way send port to send request messages to Oracle database and receive responses.
- A one-way send port to send the responses from Oracle database to a folder.

A sample orchestration for the Select operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> Set Identifier to <i>MessageIn</i> Set Type to <i>MessageInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> Set Identifier to <i>LOBPort</i> Set Type to <i>LOBPortType</i> Set Communication Pattern to <i>Request-Response</i> Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> Set Identifier to <i>ResponseOut</i> Set Type to <i>ResponseOutType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.Insert.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Insert.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Insert.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.Insert.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to Oracle E-Business Suite.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from Oracle E-Business Suite.
 - Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to Oracle E-Business Suite. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#)

To perform operations on interface tables or interface views using the Oracle E-Business adapter, you must set the right application context in which the operation is invoked. The Oracle E-Business adapter provides certain binding properties to specify the application context for any operation. You must set these binding properties on the WCF-Custom or WCF-OracleEBS port used for performing operations on interface tables.

- If the **ClientCredentialType** binding property is set to **Database**, then you must specify the following binding properties to set the applications context.

Binding Property	Value

OracleUserName	Specify the name of an Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OracleUserName binding property when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.
OraclePassword	The password for the Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OraclePassword binding property when it connects to Oracle E-Business Suite. The password is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.
OracleEBSResponsibilityName	The responsibility associated with the Oracle E-Business Suite user.

- If the **ClientCredentialType** binding property is set to **EBusiness**, you must have already specified Oracle E-Business credentials while establishing the connection. In such a case you must only specify value for the **OracleEBSResponsibilityName** binding property.

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For more information on how the adapter supports setting the application context, see [Setting Application Context](#).

Note

You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for inserting records into the AR_ARCHIVE_PURGE_INTERIM interface table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to send messages to Oracle E-Business Suite is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the Insert operation that you generated earlier. For example, the request message to select all records from the AR_ARCHIVE_PURGE_INTERIM interface table is:

```
<Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/AR/AR/AR_ARCHIVE_PURGE_INTERIM">
  <RECORDSET>
    <InsertRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecord/AR/AR_ARCHIVE_PURGE_INTERIM">
      <TRX_ID>001</TRX_ID>
      <RELATED_ID>002</RELATED_ID>
    </InsertRecord>
  </RECORDSET>
</Insert>
```

This request message will insert records into the AR_ARCHIVE_PURGE_INTERIM interface table. See [Message Schemas for Insert, Update, Delete, and Select Operations](#) for more information about the request message schema for performing basic DML operations on Oracle E-Business Suite using the Oracle E-Business adapter.

For simple data columns, like in the preceding request message, you can also use the **InlineValue** attribute. For more information about the InlineValue attribute, see the description of the Insert operation in [Operations on Interface Tables and Interface Views](#).

For example, the preceding request message with inline values will resemble the following:

```
<Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/AR/AR/AR_ARCHIVE_PURGE_INTERIM">
  <RECORDSET>
    <InsertRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecord/AR/AR_ARCHIVE_PURGE_INTERIM">
      <TRX_ID InlineValue="(Select TRX_ID FROM table_name)">001</TRX_ID>
      <RELATED_ID>002</RELATED_ID>
    </InsertRecord>
  </RECORDSET>
</Insert>
```

In this request message, the value for TRX_ID column is retrieved from another table. So, even though "001" is specified as a value for TRX_ID, the value for the SELECT statement specified for InlineValue attribute will get inserted into the table.

The orchestration consumes the message and sends it to Oracle E-Business Suite. The response from Oracle E-Business Suite is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle E-Business Suite for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<InsertResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/AR/AR/AR_ARCHIVE_PURGE_INTERIM">
  <InsertResult>1</InsertResult>
</InsertResponse>
```

The response contains the number of rows inserted into the table.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. After you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Tables with Large Data Types Using BizTalk Server

The Oracle E-Business adapter enables adapter clients to perform operations on interface tables and views with large data types such as BLOB, CLOB, NCLOB, and BFILE.

- For columns of type BLOB, CLOB, and NCLOB the adapter enables clients to read as well as update data. The adapter exposes `Read_<LOBColumnName>` and `Update_<LOBColumnName>` operations to read and update data respectively, where `<LOBColumnName>` is the name of column with large data type. If there is more than one column with large data type in a single interface table, the adapter exposes as many read and update operations for that interface table.
- For columns of type BFILE, adapter clients can only read data. The adapter exposes `Read_<LOBColumnName>` operation to read data from columns of BFILE type. If there is more than one column with large data type in a single interface table, the adapter exposes as many read operations for the interface table.

For more information about these operations, see

[Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#). For information about message schemas for performing these operations, see [Message Schemas for Special LOB Operations](#).

How to Perform Operations on Columns with Large Data Types

Performing an operation on Oracle E-Business Suite by using Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform operations on interface tables and interface views in Oracle E-Business Suite that contain large data types, these tasks are:

1. Create a BizTalk project, and generate schema for the operation (`Read_<LOBColumnName>` or `Update_<LOBColumnName>`) you want to invoke on a table or view.
2. Create messages in the BizTalk project for sending and receiving messages from the Oracle E-Business Suite.
3. Create an orchestration to invoke the operation on the interface table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

How This Topic Demonstrates Reading and Writing Data Into Columns of Large Data Types

To demonstrate reading and writing data into columns of large data types, this topic provides instructions to create an orchestration that does the following:

- Update the PHOTO column (of BLOB data type) of the CUSTOMER table.
- Read the value for the PHOTO column for the updated record.

This orchestration is designed in such a way that you only provide the request message for the update operation at run-time. The message for the read operation will be constructed within the operation.

Note

The orchestration in this topic reads and updates data from the CUSTOMER table, which is a base database table created by running the scripts provided with the samples. You must perform similar procedures as described in this topic to perform read or update operations on any interface table or interface view.

Generating Schema

This topic demonstrates how to perform basic read and update operations on a PHOTO column (of BLOB data type) in a CUSTOMER table. This table is created by running the scripts provided with the samples.

To demonstrate how to read and write data to a column of large data type, schema is generated for the **Update_PHOTO** and **Read_PHOTO** operations for the CUSTOMER table. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) for more information about how to generate schemas.

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration, and link them to schemas that you generated in the previous step.

In this orchestration you must create four messages—one receive-response set for the **Update_PHOTO** operation and the other receive-response set for the **Read_PHOTO** operation.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
Identifier	Type UpdateMessage
Message Type	From the drop-down list, expand Schemas , and then select <i>LOBOperations.OracleEBSBinding.Update_PHOTO</i> , where LOBOperations is the name of your BizTalk project. OracleEBSBindingSchema is the schema generated for invoking the Update_PHOTO operation on CUSTOMER table.

6. Repeat step 3 to create three new message. In the **Properties** pane for the new message, do the following:

Set Identifier to	Set Message Type to
UpdateResponse	<i>LOBOperations.OracleEBSBinding.Update_PHOTOResponse</i>
ReadMessage	<i>LOBOperations.OracleEBSBinding1.Read_PHOTO</i>
ReadResponse	<i>LOBOperations.OracleEBSBinding1.Read_PHOTOResponse</i>

Setting up the Orchestration

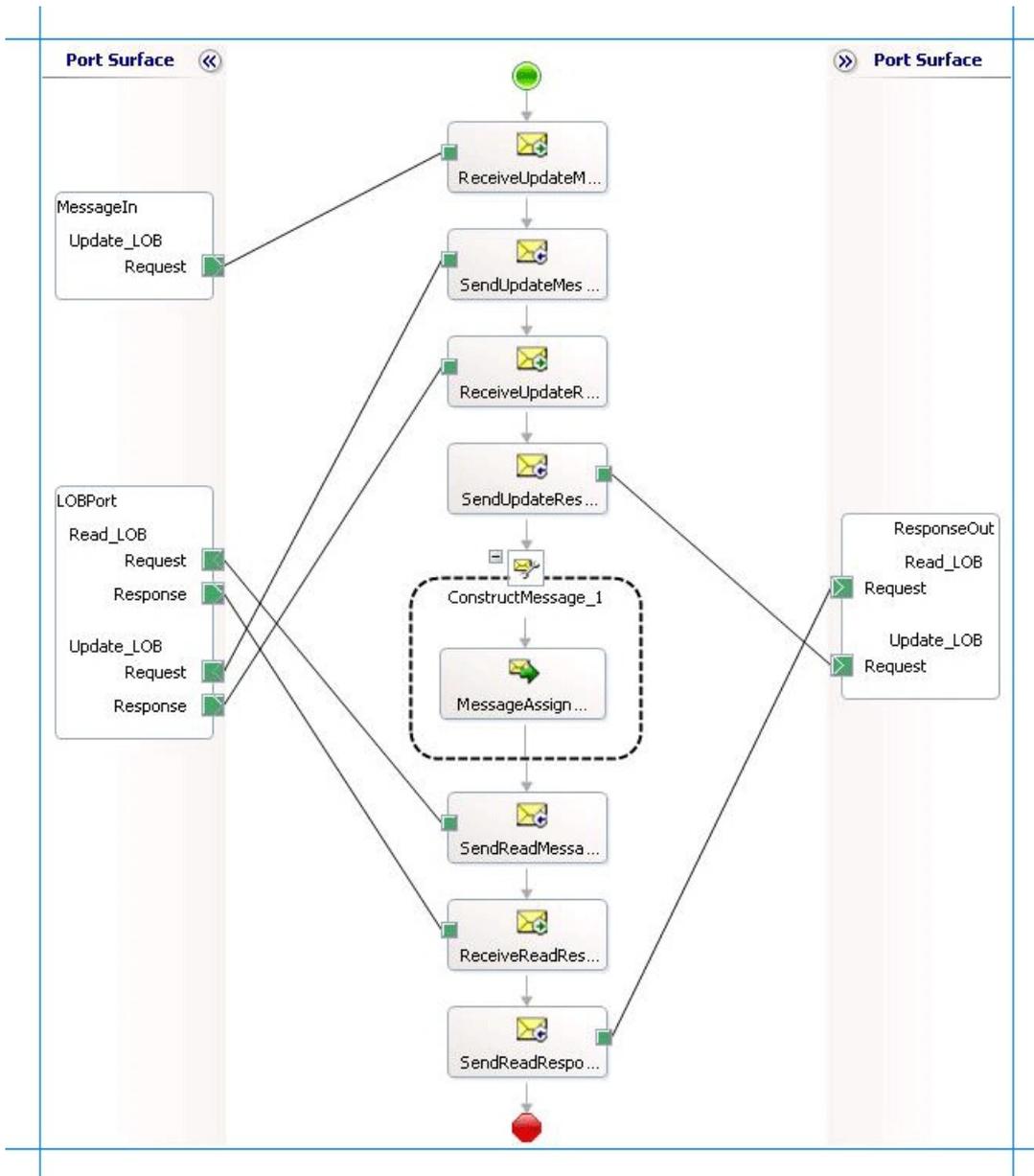
In this orchestration, the adapter receives a request message to perform an Update_PHOTO operation on a CUSTOMER table. The notification message is received at a FILE location. The adapter consumes this message and passes it on to Oracle database. The response from Oracle database is saved at another location. Once the response is received, the orchestration constructs a message to invoke the Read_PHOTO operation, which reads the value of the PHOTO column updated by the Update_PHOTO operation. The response for this message is also received at the same FILE location.

So, your orchestration must contain the following:

- A FILE receive port to drop a request message for **Update_PHOTO** operation.
- A two-way WCF-Custom or WCF-OracleEBS send port to send messages to execute the **Update_PHOTO** operation.

- A two-way WCF-Custom or WCF-OracleEBS send port to send messages to execute the **Read_PHOTO** operation. You can also perform both **Read_PHOTO** and **Update_PHOTO** using the same WCF-Custom or WCF-OracleEBS send port. In this topic, you will use a single send port for both the operations.
- A **Construct Message** shape to construct messages within the orchestration.
- A FILE send port to save the response messages for **Update_PHOTO** and **Read_PHOTO** operations.
- Receive and send shapes.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveUpdateMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveUpdateMessage</i> • Set Activate to <i>True</i>

SendUpdateMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendUpdateMessage</i>
ReceiveUpdateResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveUpdateResponse</i>
SendUpdateResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>SendUpdateResponse</i>
SendReadMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendReadMessage</i>
ReceiveReadResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveReadResponse</i>
SaveReadResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveReadResponse</i>

Adding Construct Message Shape

You can use the **Construct Message** shape to generate a request message within the orchestration to execute the **Read_PHOTO** operation. To do so, you must add a **Construct Message** shape and within that a **Message Assignment** shape to your orchestration. For this example, the **Message Assignment** shape invokes code that generates a message that is sent to Oracle E-Business Suite to execute the **Read_PHOTO** operation. The **Message Assignment** shape also sets the action for the message to be sent to Oracle E-Business Suite.

For the construct message shape, set the **Message Constructed** property to **ReadMessage**.

The code to generate the response could be part of the same Visual Studio solution as your BizTalk project. A sample code for generating a response message looks like this.

```
namespace MessageCreator
{
    public class MessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;

        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\MessageIn";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOption.To
                pDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }
            //Create Message From XML
            Message = new XmlDocument();
            Message.PreserveWhitespace = true;
            Message.Load(ResponseDoc);
            return Message;
        }
    }
}
```

For the above code excerpt to be able to generate a request message, you must have an XML request message (for the **Read_PHOTO** operation) in the location specified for the `XmlFileLocation` variable.

Note

After you build the project, MessageCreator.dll will be created in the project directory. You must add this DLL to the global assembly cache (GAC). Also, you must add the MessageCreator.dll as a reference in the BizTalk project.

Add the following expression to invoke this code from the **Message Assignment** shape and to set the action for message. To add an expression, double-click the **Message Assignment** shape to open the Expression Editor.

```
ReadMessage = MessageCreator.MessageCreator.XMLMessageCreator();
ReadMessage(WCF.Action) = "Tables/ReadLOB/SCOTT/CUSTOMER/PHOTO ";
```

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i> • Create an operation <i>Read_LOB</i>. This operation is used for messages to read values from large data type columns. • Create an operation <i>Update_LOB</i>. This operation is used for messages to update values in large data type columns.
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i> • Create an operation <i>Read_LOB</i>. This operation is used for messages to read values from large data type columns. • Create an operation <i>Update_LOB</i>. This operation is used for messages to update values in large data type columns.

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to

link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveUpdateMessage	<ul style="list-style-type: none"> • Set Message to <i>UpdateMessage</i> • Set Operation to <i>MessageIn.Update_LOB.Request</i>
SendUpdateMessage	<ul style="list-style-type: none"> • Set Message to <i>UpdateMessage</i> • Set Operation to <i>LOBPort.Update_LOB.Request</i>
ReceiveUpdateResponse	<ul style="list-style-type: none"> • Set Message to <i>UpdateResponse</i> • Set Operation to <i>LOBPort.Update_LOB.Response</i>
SendUpdateResponse	<ul style="list-style-type: none"> • Set Message to <i>UpdateResponse</i> • Set Operation to <i>ResponseOut.Update_LOB.Request</i>
SendReadMessage	<ul style="list-style-type: none"> • Set Message to <i>ReadMessage</i> • Set Operation to <i>LOBPort.Read_LOB.Request</i>
ReceiveReadResponse	<ul style="list-style-type: none"> • Set Message to <i>ReadResponse</i> • Set Operation to <i>LOBPort.Read_LOB.Response</i>
SendReadResponse	<ul style="list-style-type: none"> • Set Message to <i>ReadResponse</i> • Set Operation to <i>ResponseOut.Read_LOB.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding FILE port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to Oracle database.
 - Define a location on the hard disk and a corresponding FILE port where the BizTalk orchestration will drop the

response message containing the response from Oracle database.

- Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to Oracle database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). You must make the following considerations while configuring the WCF-Custom or WCF-OracleEBS send port.
 - An Update_<LOBColumnName> operation must be performed as part of transaction. To ensure this, the **UseAmbientTransaction** binding property must be set to **True**.
 - Because the WCF-Custom or WCF-OracleEBS send port sends and receives messages conforming to more than one schema and performs two operations, you must set dynamic action for both the operations. For more information about actions, see [Specifying SOAP Action](#). For this orchestration, the action should be set as follows:

```
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="Update_LOB" Action="Tables/UpdateBlob/SCOTT/CUSTOMER/PHOTO" />
  <Operation Name="Read_LOB" Action="Tables/ReadLOB/SCOTT/CUSTOMER/PHOTO" />
</BtsActionMapping>
```

◆ Important

Note that the operation name in a dynamic action must be same as the operation name you specified on the logical ports while creating the BizTalk orchestration.

📌 Note

To perform operations on interface tables or interface views you must also set the application context. For more information on how the adapter supports setting the application context, see [Setting Application Context](#). You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

📌 Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

Before starting the BizTalk orchestration, make sure the request XML to invoke the **Read_PHOTO** operation is available at C:\TestLocation\MessageIn. The request XML must resemble the following:

```
<Read_PHOTO xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/CUSTOMER">
  <FILTER>WHERE NAME='Mindy Martin'</FILTER>
</Read_PHOTO>
```

📌 Note

The request message has a filter on the specific name because in the request message for **Update_PHOTO** operation, the value for the PHOTO column is updated for the same name. So, the read operation will read the same value that you insert using the update operation.

You must now start the BizTalk application for reading and writing values of large data types from an Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to send messages to Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you start the application you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the **Update_PHOTO** operation you generated earlier. For example, a request message that updates the PHOTO column of the CUSTOMER table resembles the following:

```
<Update_PHOTO xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/CUSTOMER">
  <FILTER>WHERE Name='Mindy Martin'</FILTER>
  <DATA>U2FtcGx1IERhdGE=</DATA>
</Update_PHOTO>
```

Note

While updating BLOB columns, the DATA element must always contain a base64 encoded value. For CLOB and NCLOB, the DATA element can have string values.

The preceding request message updates the value in the PHOTO column for the record matching the WHERE clause. See Message Schemas for Operations on Large Data Types for more information about the request message schema for performing operations on large data types using the Oracle E-Business adapter.

The orchestration consumes the message and sends it to Oracle database. The response from Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the preceding request message resembles the following:

```
<?xml version="1.0" encoding="utf-8"?>
<Update_PHOTOResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/CUSTOMER" />
```

The orchestration now constructs a request message for the **Read_PHOTO** operation by using the request message available at C:\TestLocation\MessageIn. The request message is sent to Oracle database and the response is saved at the same FILE location. The response for the read operation on PHOTO column resembles the following:

```
<?xml version="1.0" encoding="utf-8"?>
<Read_PHOTOResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/CUSTOMER">
  <Read_PHOTOResult>U2FtcGx1IERhdGE=</Read_PHOTOResult>
</Read_PHOTOResponse>
```

Note

Note that the response contains the same value for the PHOTO column that you passed in the **Update_PHOTO** operation.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see

[Reusing Adapter Bindings.](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Composite Operations on Oracle Database by Using BizTalk Server

The Oracle E-Business adapter enables adapter clients to perform composite operations on Oracle database. A composite operation can include:

- Insert, Update, Delete, and Select operations on database tables. Select operation on database views.
- Insert, Update, Delete, and Select operations on interface tables. Select operation on interface views.
- Stored procedures and functions, inside or outside a package.

A single composite operation can have any number of these operations, in any order. For example, you can have two inserts followed by a delete, and finally a stored procedure execution. Also, you can have different operations targeting different database tables or views. For more information about how the adapter supports composite operations, see [Support for Composite Operations](#). For information about the structure of the SOAP message for composite operations, see [Message Schemas for the Composite Operation](#).

How to Perform Composite Operations on Oracle Database?

Performing an operation on Oracle database using Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform composite operations on Oracle database, these tasks are:

1. Create a BizTalk project in Visual Studio and generate schema for all the operations you want to invoke.
2. Manually create a schema file that includes references to all the schemas you generated in the previous step.
3. Create messages in the BizTalk project for sending and receiving messages from Oracle database. These messages must conform to the request and response schema you created in the previous step.
4. Create an orchestration to invoke the composite operation on Oracle database.
5. Build and deploy the BizTalk project.
6. Configure the BizTalk application by creating physical send and receive ports.
7. Start the BizTalk application.

This topic provides instructions on how to perform these tasks.

Generating Schema

In this topic, to demonstrate how to perform composite operations, we will perform the following tasks in the same order:

- Insert record into the ACCOUNTACTIVITY table.
- Retrieve all the records in the ACCOUNTACTIVITY table by invoking the GET_ALL_ACTIVITY procedure within the ACCOUNT_PKG package.
- Delete the record from the ACCOUNTACTIVITY table.

Run the scripts provided with the samples to create the ACCOUNTACTIVITY table. For more information about the samples, see [Samples](#).

You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) for more information about how to generate

schemas.

Creating a Composite Schema Definition

You must now create a composite schema in the Visual Studio BizTalk project that references the schemas you created for the individual operations. Perform the following steps to create a composite schema definition.

To add a composite schema definition

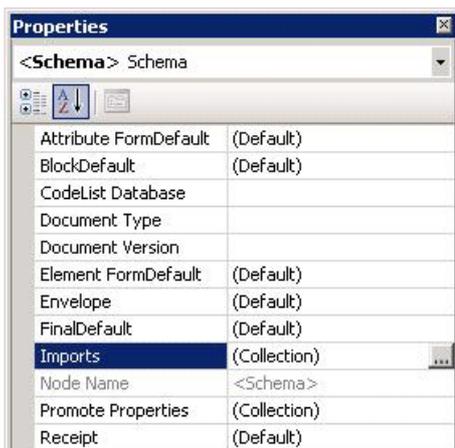
1. Add a schema file to the BizTalk project in Visual Studio. Right-click the solution name, point to **Add**, and then click **New Item**. In the **Add New Item** dialog box, from the **Categories** box, click **Schema Files**. From the **Templates** box, click **Schema**. Specify a name for the schema file and click **OK**.

For this example, specify the schema file name as **CompositeSchema.xsd**.

2. Add references to the schema generated for the different operations that you want to perform. In this example, the different schemas generated for operations are:
 - OracleEBSBinding.xsd, for Insert and Delete operations on ACCOUNTACTIVITY table.
 - OracleEBSBinding2.xsd, for the GET_ALL_ACTIVITY procedure.

To add references:

- a. Right-click the root **<Schema>** node in the CompositeSchema.xsd, and click **Properties**.
- b. In the **Property** box, click the ellipsis button (...) against the **Imports** property.



- c. In the **Imports** dialog box, from the **Import new schema as** list, select **XSD Import**, and then click **Add**.
- d. In the **BizTalk Type Picker** dialog box, expand the BizTalk project name node, expand **Schemas**, and then select the schema you want to import. For this example, select **<BizTalk_project_name>.OracleEBSBinding.xsd**. Click **OK**.

Repeat this step to import **<BizTalk_project_name>.OracleEBSBinding2.xsd** too.

- e. In the **Imports** dialog box, click **OK**.
3. Add two child nodes to the root schema node. One child node corresponds to the request schema for performing the composite operation. The other child node corresponds to the response schema. The node that corresponds to the request schema can have any name. The node that corresponds to the response schema must be called **<request_schema_node>Response**. For this example, we will call the request schema node as **Request**. So, the response schema node is called **RequestResponse**.

Note

By default, a **Root** node is also added to a new schema file. You can rename the **Root** node to **Request**. To rename a node, right-click the node name and click **Rename**.

To add a node under the **<Schema>** node:

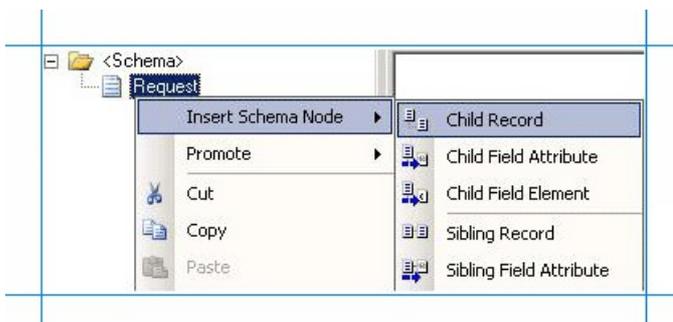
- a. Right-click the **<Schema>** node, point to **Insert Schema Node**, and click **Child Record**.
 - b. Rename the new node to **RequestResponse**.
4. Add child nodes under the **Request** node that correspond to the request schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:
- Insert and Delete operations on the ACCOUNTACTIVITY table.
 - GET_ALL_ACTIVITY procedure.

Important

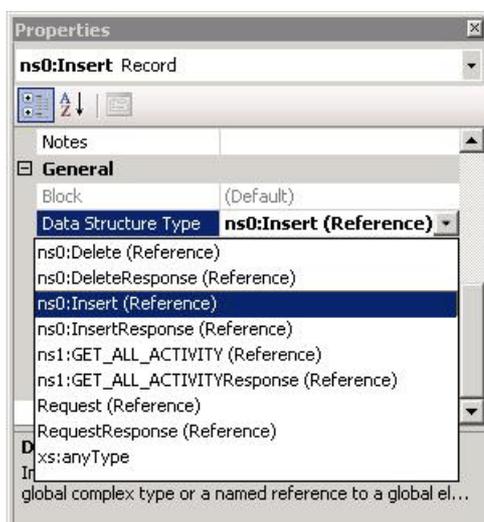
You must add the nodes in the same order in which you want to perform the operations. For example, if you want to insert a record, then execute a stored procedure, and then delete a record you must first add a node for the Insert operation, followed by a node for the stored procedure, and finally a node for the Delete operation.

To add child nodes to the **Request** node:

- a. Right-click the **Request** node, point to **Insert Schema Node**, and then click **Child Record**.



- b. Rename the record to correspond to a request schema for an operation that you perform as part of the composite operation. For example, rename the node to "Insert".
- c. Map the **Insert** node to the request schema for the Insert operation on the ACCOUNTACTIVITY table. To do so, right-click the **Insert** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **Insert (Reference)**.



- d. Repeat these steps to add nodes for the request schemas for GET_ALL_ACTIVITY stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following

table.

Node name	Mapped to schema
GET_ALL_ACTIVITY	GET_ALL_ACTIVITY (Reference)
Delete	Delete (Reference)

5. Add child nodes under the **RequestResponse** node that correspond to the response schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:

- Insert and Delete operations on the ACCOUNTACTIVITY table.
- GET_ALL_ACTIVITY stored procedure.

Important

You must add the child nodes in the same order as the child nodes under the **Request** node.

To add child nodes to the **RequestResponse** node:

- a. Right-click the **RequestResponse** node, point to **Insert Schema Node**, and click **Child Record**.
- b. Rename the record to correspond to a response schema for an operation that you perform as part of the composite operation. For example, rename the node to "InsertResponse".
- c. Map the **InsertResponse** node to the response schema for the Insert operation on the ACCOUNTACTIVITY table. To do so, right-click the **InsertResponse** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **InsertResponse (Reference)**.
- d. Repeat these steps to add nodes for the response schemas for the GET_ALL_ACTIVITY stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following table.

Node name	Mapped to schema
GET_ALL_ACTIVITYResponse	GET_ALL_ACTIVITYResponse (Reference)
DeleteResponse	DeleteResponse (Reference)

6. Save the **CompositeSchema.xsd** file.

Defining Messages and Message Types

The composite schema that you created in the last step describes the "types" required for the messages in an orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schema you created in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project in Visual Studio. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

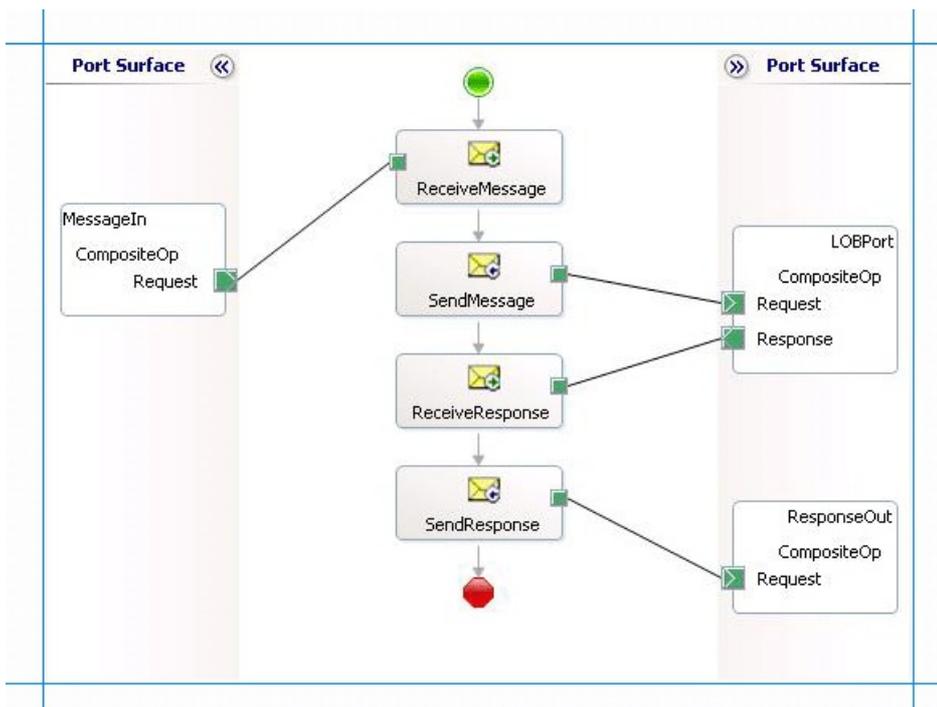
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>Composite_Op.CompositeSchema.Request</i> , where <i>Composite_Op</i> is the name of your BizTalk project. <i>CompositeSchema</i> is the schema you created manually for the composite operations.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>Composite_Op.CompositeSchema.RequestResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing composite operations on Oracle database. In this orchestration, you drop a request message at a defined receive location. The request message must conform to the composite schema you created earlier. The Oracle E-Business adapter consumes this message and passes it on to Oracle database. The response from Oracle database is saved to another location. You must include Send and Receive shapes to send messages to Oracle database and receive responses, respectively. A basic orchestration for performing composite operations resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>

ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.CompositeOp.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.CompositeOp.Request</i>

ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.CompositeOp.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.CompositeOp.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from Oracle database.
 - Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to Oracle database. Because the operations that are being as part of the composite operation are executed in a single transaction, make sure the **UseAmbientTransaction** binding property is set to **True**.

You must also specify the action in the send port. The action for a composite operation is "CompositeOperation". For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). For more information about how to specify actions for ports, see [Specifying SOAP Action](#).

◆ Important

As part of composite operations, if you are executing operations on objects, for example stored procedures, functions, interface tables, or interface views, which belong to an Oracle E-Business Suite application, you must set the application context by specifying the requisite binding properties. For more information about setting the application context, see [Setting Application Context](#).

You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

📌 Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#). If you import this binding file, the action on the send port is set to a dynamic action involving all the operations you selected in the Consume Adapter Service Add-in while generating the schema. For a composite operation, you must replace the dynamic action with "CompositeOperation".

Starting the Application

You must start the BizTalk application for performing composite operations on Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to send messages to Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the composite operations you created earlier. For example, a request message that inserts a record in the ACCOUNTACTIVITY table, invokes the GET_ALL_ACTIVITY stored procedure, and deletes a record from the ACCOUNTACTIVITY table is:

```
<Request xmlns="http://Composite_Op.CompositeSchema">
  <Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/ACCOUNTACTIVITY">
    <RECORDSET>
      <InsertRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecord/SCOTT/ACCOUNTACTIVITY">
        <TID InlineValue="tid_seq.nextval"></TID>
        <ACCOUNT>100001</ACCOUNT>
        <AMOUNT>1500</AMOUNT>
        <DESCRIPTION></DESCRIPTION>
        <TRANSDATE InlineValue="sysdate">1999-05-31T13:20:00</TRANSDATE>
        <PROCESSED>n</PROCESSED>
      </InsertRecord>
    </RECORDSET>
  </Insert>
  <GET_ALL_ACTIVITY xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/SCOTT/ACCOUNT_PKG" />
  <Delete xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/ACCOUNTACTIVITY">
    <FILTER>WHERE AMOUNT = 1500</FILTER>
  </Delete>
</Request>
```

The preceding request message first inserts a record and then invokes the GET_ALL_ACTIVITY procedure to get all the records in the ACCOUNTACTIVITY table. Then, the inserted record is deleted by specifying a FILTER clause. See Message Schemas for the composite operations for more information about the request message schema for performing composite operations on Oracle database using the Oracle E-Business adapter.

Note

In the preceding message, the excerpt for the Insert operation uses the "InlineValue" attribute. For more information about the "InlineValue" attribute see the schema description for Insert operation in [Message Schemas for Insert, Update, Delete, and Select Operations](#).

The orchestration consumes the message and sends it to Oracle database. The response from Oracle database is saved at the other FILE location defined as part of the orchestration. For example, the response from Oracle database for the preceding request message resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<RequestResponse xmlns="http://Composite_Op.CompositeSchema">
  <InsertResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/ACCOUNTACTIVITY">
    <InsertResult>1</InsertResult>
  </InsertResponse>
  <GET_ALL_ACTIVITYResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/SCOTT/ACCOUNT_PKG">
    <ALLRECS>
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                    <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                    <xs:element minOccurs="0" name="AMOUNT" type="xs:decimal" />
                    <xs:element minOccurs="0" name="DESCRIPTION" type="xs:string" />
                    <xs:element minOccurs="0" name="TRANSDATE" type="xs:dateTime" />
                    <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
      <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
        <NewDataSet xmlns="">
          <NewTable>
            .....
            .....
          </NewTable>
          .....
          .....
          <NewTable>
            <TID>10</TID>
            <ACCOUNT>100001</ACCOUNT>
            <AMOUNT>1000</AMOUNT>
            <TRANSDATE>2008-07-28T21:39:57</TRANSDATE>
            <PROCESSED>n</PROCESSED>
          </NewTable>
        </NewDataSet>
      </diffgr:diffgram>
    </ALLRECS>
  </GET_ALL_ACTIVITYResponse>
  <DeleteResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/SCOTT/ACCOUNTACTIVITY">
    <DeleteResult>1</DeleteResult>
  </DeleteResponse>
</RequestResponse>
```

The preceding response contains multiple result sets corresponding the different operations performed as part of the

composite operation. For example, the `InsertResult` element contains '1', indicating the number of rows inserted by the Insert operation. Similarly, the `DeleteResult` element contains '1', indicating the number of rows deleted by the Delete operation.

◆ Important

If you experience time-out issues while executing a composite operation then it could be because the number of connections is less than the number of operations in a composite operation involving:

- Stored procedures containing BFILE, BLOB, CLOB, NCLOB, and REF CURSOR as OUT or IN OUT parameters.
- Select operation.

To resolve this issue, you must ensure that if there are "n" number of such operations in a composite operation, the value specified for the **MinPoolSize** binding property is "n+1" or greater. For more information about the **MinPoolSize** binding property, see [Working with BizTalk Adapter in Oracle E-Business Suite Binding Properties](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Concurrent Programs in Oracle E-Business Suite Using BizTalk Server

Oracle E-Business Suite exposes concurrent programs that you can execute to perform specific operations on Oracle applications. Each Oracle application has a set of standard concurrent programs (that are same across all operations) and certain concurrent programs that are specific to an Oracle application. The Oracle E-Business adapter exposes all concurrent programs as operations that adapter clients can invoke. For more information on how the adapter supports concurrent programs, see [Operations on Concurrent Programs](#). For information about the structure of SOAP messages for invoking concurrent programs, see [Message Schemas for Concurrent Programs](#).

Note

For the concurrent programs that do not expose their metadata, the Oracle E-Business adapter exposes 100 optional parameters for each of these concurrent programs. To invoke these concurrent programs successfully, the user must consult the Oracle E-Business Suite documentation to figure out the parameters for a concurrent program that require a value, and then specify them. An example of such a concurrent program is **Journal Import** (actual name: **GLLEZL**) in the **General Ledger** application.

Prerequisites

You must have completed the steps in [Before You Develop BizTalk Applications](#).

How to Invoke Concurrent Programs in Oracle Applications

Performing an operation on Oracle E-Business Suite using the Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a concurrent program, these tasks are:

- Create a BizTalk project, and generate schema for the concurrent program you want to invoke.
- Create messages in the BizTalk project for sending and receiving messages to and from Oracle E-Business Suite.
- Create an orchestration to invoke the concurrent program.
- Build and deploy the BizTalk project.
- Configure the BizTalk application by creating physical send and receive ports.
- Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke the **Customer Interface** concurrent program from the **Receivables** application. This application is available with the default Oracle E-Business Suite application. This concurrent program returns a request ID. To check the status of the concurrent program, we execute the **Get_Status** concurrent program by passing the request ID received in the response of the **Customer Interface** concurrent program.

In this topic, we generate schema for both the **Customer Interface** and **Get_Status** concurrent programs. For more information about how to generate schema, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration, and link them to schemas that you generated in the previous step.

In this orchestration you must create four messages—one receive-response set to invoke the **Customer Interface** concurrent program and the other receive-response set to invoke the **Get_Status** concurrent program.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>ConcurrentProgram.OracleEBSBindingSchema.RACUST</i> , where <i>ConcurrentProgram</i> is the name of your BizTalk project. <i>OracleEBSBindingSchema</i> is the schema generated for invoking the Customer Interface concurrent program.  Note RACUST is the actual name of the Customer Interface concurrent program. While the Consume Adapter Service Add-in displays the friendly name (Customer Interface), the schema contains the actual name of the concurrent program.

6. Repeat step 3 to create three new message. In the **Properties** pane for the new message, do the following:

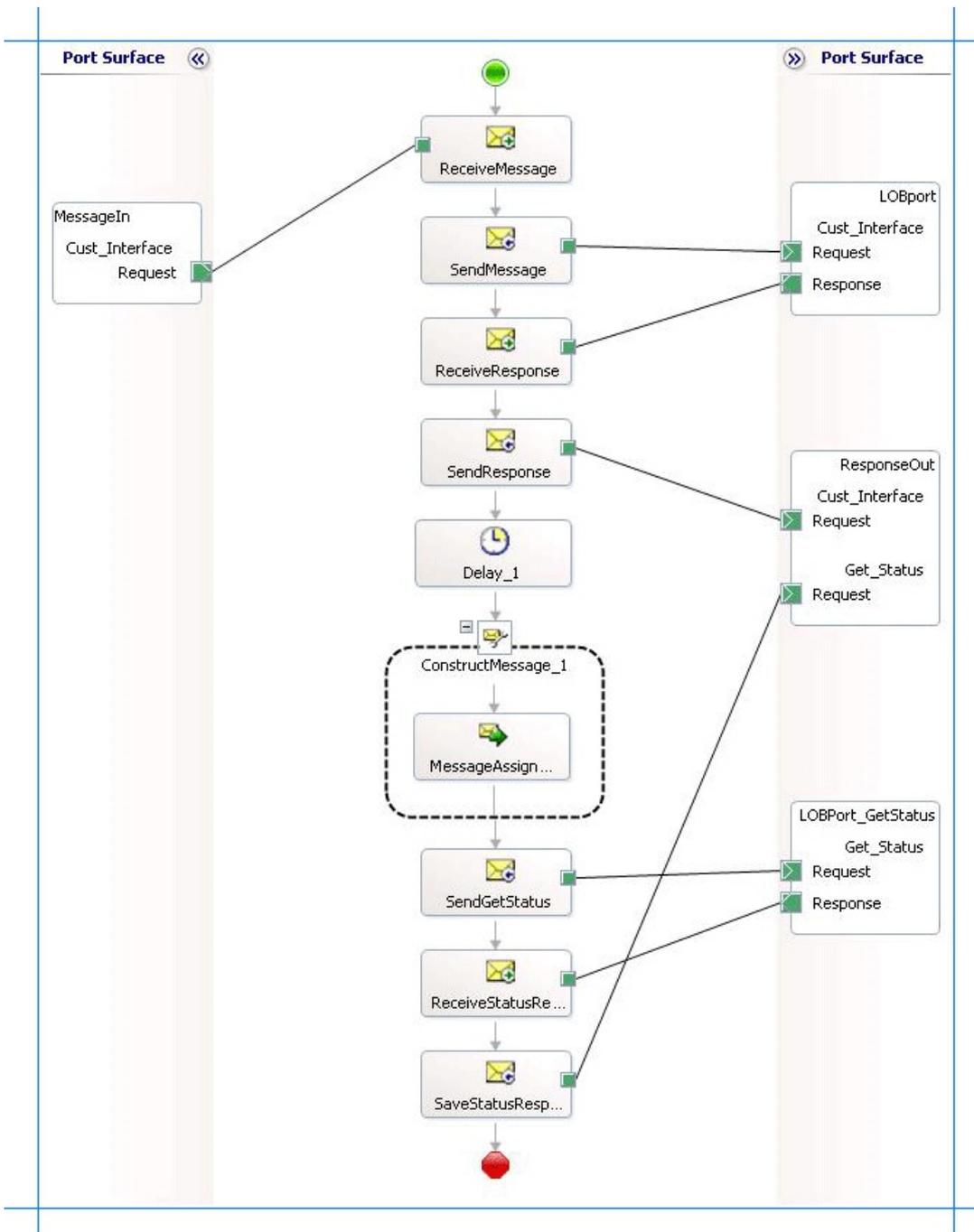
Set Identifier to	Set Message Type to
Response	<i>ConcurrentProgram.OracleEBSBindingSchema.RACUSTResponse</i>
Get_StatusRequest	<i>ConcurrentProgram.OracleEBSBindingSchema1.GetStatusForConcurrentProgram</i>
Get_StatusResponse	<i>ConcurrentProgram.OracleEBSBindingSchema1.GetStatusForConcurrentProgramResponse</i>

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking concurrent programs in Oracle E-Business Suite. In this orchestration, you will be dropping a request message at a defined receive location. The orchestration consumes this message and passes it on to Oracle E-Business Suite to invoke the **Customer Interface** concurrent program. The response for the concurrent program is received from Oracle and is saved at another location. The response message contains a request ID. The orchestration includes a **Construct Message** shape to extract the request ID from the response and construct a message that conforms to the schema of the **Get_Status** concurrent program. The message to invoke the **Get_Status** concurrent program is sent to the Oracle E-Business Suite with the request ID as a parameter. You must include send and receive shapes, message construction shapes, and ports to send messages to Oracle and receive responses.

Typically, the **Customer Interface** concurrent program will take some time to execute, so you need to wait before executing the **Get_Status** concurrent. You can automate this by adding a **Delay** shape.

A sample orchestration resembles the following:



Adding Message Shapes

Specify the following properties for each of the message shapes. The names listed in the **Shape** column correspond to the message shapes as displayed in the preceding orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

SendGetStatus	Send	<ul style="list-style-type: none"> • Set Name to <i>SendGetStatus</i>
ReceiveStatusResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveStatusResponse</i> • Set Activate to <i>False</i>
SaveStatusResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveStatusResponse</i>

Adding a Delay Shape

If you want the orchestration to wait between invoking the **Customer Interface** and **Get_Status** concurrent programs, you must add a **Delay** shape to the orchestration. You must add a **Delay** shape after the orchestration copies the response for the **Customer Interface** concurrent program to a FILE send port. So, you must add a **Delay** shape after the **SendResponse** shape.

Within the **Delay** shape, you can specify the time interval for which the orchestration must wait before proceeding by adding the following code to the Expression Editor for the **Delay** shape:

```
new System.TimeSpan(0,2,0)
```

By adding this code, the orchestration will wait for two minutes before proceeding. For more information about how to configure the **Delay** shape, see <http://go.microsoft.com/fwlink/?LinkId=122405>.

Adding the Construct Message Shape

The response from the Oracle E-Business Suite for the **Customer Interface** concurrent program contains a request ID. To get the status of the concurrent program, you must pass the same request ID as a parameter to the **Get_Status** concurrent program. To do so, in the orchestration, you must include a **Construct Message** shape, and within that a **Message Assignment** shape. The purpose of the **Construct Message** shape is:

- To extract the request ID from the response that is received for the **Customer Interface** concurrent program.
- To construct a message that conforms to the message schema for the **Get_Status** concurrent program.

For the **Construct Message** shape, set the **Message Constructed** property to **Get_StatusRequest**.

For the **Message Assignment** shape, add the below. Before adding the code, you must have:

```
XmlDoc = new System.Xml.XmlDocument();
XmlDoc.LoadXml("<GetStatusForConcurrentProgram xmlns='http://schemas.microsoft.com/OracleEBS/2008/05/ConcurrentPrograms/AR'><RequestId /></GetStatusForConcurrentProgram>");
Get_StatusRequest = XmlDoc;
Get_StatusRequest.RequestId = xpath(Response,"string(/*[local-name()='RACUSTResponse']/*[local-name()='RACUSTResult']/text())");
```

Adding Ports

To configure the ports, you specify the properties listed in the table below for each of the logical ports. The names listed in the *Port* column correspond to the names of the ports displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i> • Create an operation <i>Cust_Interface</i>. This operation is used for the Customer Interface concurrent program. • Create an operation <i>Get_Status</i>. This operation is used for the Get_Status concurrent program.
LOBPort_GetStatus	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort_GetStatus</i> • Set Type to <i>LOBPort_GetStatusType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the property values to specify messages for action shapes and linking them to the ports. The names listed in the **Shape** column correspond to the names of the message shapes as displayed in the orchestration diagram.

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and then deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.Cust_Interface.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBport.Cust_Interface.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBport.Cust_Interface.Response</i>

SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.Cust_Interface.Request</i>
SendGetStatus	<ul style="list-style-type: none"> • Set Message to <i>Get_StatusRequest</i> • Set Operation to <i>LOBPort_GetStatus.Get_Status.Request</i>
ReceiveStatusResponse	<ul style="list-style-type: none"> • Set Message to <i>Get_StatusResponse</i> • Set Operation to <i>LOBPort_GetStatus.Get_Status.Response</i>
SaveStatusResponse	<ul style="list-style-type: none"> • Set Message to <i>Get_StatusResponse</i> • Set Operation to <i>ResponseOut.Get_Status.Request</i>

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle E-Business Suite.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from Oracle E-Business Suite.
- Define two physical WCF-Custom or WCF-OracleEBS send ports—one to send messages to Oracle E-Business Suite to execute the **Customer Interface** concurrent program and the other to execute the **Get_Status** concurrent program. You must also specify the action in the send ports. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).

To invoke concurrent programs using the Oracle E-Business adapter, you must set the right application context in which the operation is invoked. The Oracle E-Business adapter provides certain binding properties to specify the application context for any operation. You must set these binding properties on the WCF-Custom or WCF-OracleEBS port used for invoking concurrent programs.

- If the **ClientCredentialType** binding property is set to **Database**, then you must specify the following binding properties to set the applications context.

Binding Property	Value

OracleUserName	Specify the name of an Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OracleUserName binding property when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.
OraclePassword	The password for the Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OraclePassword binding property when it connects to Oracle E-Business Suite. The password is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.
OracleEBSResponsibilityName	The responsibility associated with the Oracle E-Business Suite user.

- If the **ClientCredentialType** binding property is set to **EBusiness**, you must have already specified Oracle E-Business credentials while establishing the connection. In such a case you must only specify value for the **OracleEBSResponsibilityName** binding property.

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For more information on how the adapter supports setting the application context, see [Setting Application Context](#).

Note

You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application before invoking the concurrent programs. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to invoke the **Customer Interface** concurrent program is running.

- The WCF-Custom or WCF-OracleEBS send port to invoke the **Get_Status** concurrent program is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message that conforms to the schema for invoking the **Customer Interface** concurrent program. For example, the request message to invoke the **Customer Interface** concurrent program is:

```
<RACUST xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ConcurrentPrograms/AR">
  <Description>Customer Interface Program</Description>
  <StartTime></StartTime>
  <CREATE_RECIPROCAL_CUSTOMER>Yes</CREATE_RECIPROCAL_CUSTOMER>
  <ORG_ID>203</ORG_ID>
</RACUST>
```

Note

The request message for invoking a concurrent program requires some optional parameters such as SetOptions, SetPrintOptions, and SetRepeatOptions. The request message provided here does not contain these optional parameters. For information on the complete request message, including optional parameters, see [Message Schemas for Concurrent Programs](#).

The orchestration consumes the message, passes it on to the Oracle E-Business Suite, and receives a response. The response message is saved at the other file location specified as part of the orchestration. The response for the Customer Interface concurrent program resembles the following:

```
<?xml version="1.0" encoding="utf-8"?>
<RACUSTResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ConcurrentPrograms/AR">
  <RACUSTResult>2794708</RACUSTResult>
</RACUSTResponse>
```

The response from the Oracle E-Business Suite contains a request ID. The orchestration extracts the request ID from the response message, constructs a message to invoke the **Get_Status** concurrent program, and passes it to the Oracle E-Business Suite to execute the **Get_Status** concurrent program. After the response is received for the **Get_Status** concurrent program, it is copied to the same file location as the first response. The response for the **Get_Status** concurrent program resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<GetStatusForConcurrentProgramResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ConcurrentPrograms/AR">
  <GetStatusForConcurrentProgramResult>true</GetStatusForConcurrentProgramResult>
  <Phase>Pending</Phase>
  <Status>Standby</Status>
  <DevPhase>PENDING</DevPhase>
  <DevStatus>STANDBY</DevStatus>
  <Message>null</Message>
</GetStatusForConcurrentProgramResponse>
```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. After you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Request Sets in Oracle E-Business Suite Using BizTalk Server

Microsoft BizTalk Adapter for Oracle E-Business Suite enables you to execute request sets in Oracle E-Business Suite. Request sets are divided into one or more stages, and each stage contains a set of reports and concurrent programs. For more information on how the adapter supports request sets, see [Operations on Request Sets](#). For information about the structure of SOAP messages for invoking request sets, see [Message Schemas for Request Sets](#).

Prerequisites

You must have completed the steps in [Before You Develop BizTalk Applications](#).

How to Invoke Request Sets in Oracle E-Business Suite

Performing an operation on Oracle E-Business Suite using the Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a request set, these tasks are:

- Create a BizTalk project, and generate schema for the request set you want to invoke.
- Create messages in the BizTalk project for sending and receiving messages to and from Oracle E-Business Suite.
- Create an orchestration to invoke the request set.
- Build and deploy the BizTalk project.
- Configure the BizTalk application by creating physical send and receive ports.
- Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke a request set, by invoking the **Function Security Reports** (friendly name) request set from the **Application Object Library** application. The actual name of the request set is **FNDRSSUB43**. This request set is available with the default Oracle E-Business Suite application. This request set returns a request ID.

Hence, in this topic, we generate schema for the **FNDRSSUB43** request set. See [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration, and link them to schemas that you generated in the previous step.

In this orchestration you must create two messages—one to send message to invoke the request set and another to receive a response for the request set.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>RequestSet.OracleEBSBindingRequestSets_FND.FNDRSSUB43</i> , where RequestSet is the name of your BizTalk project. OracleEBSBindingRequestSets is the schema generated for invoking the FNDRSSUB43 request set.

6. Repeat step 3 to create a message. In the **Properties** pane for the new message, do the following:

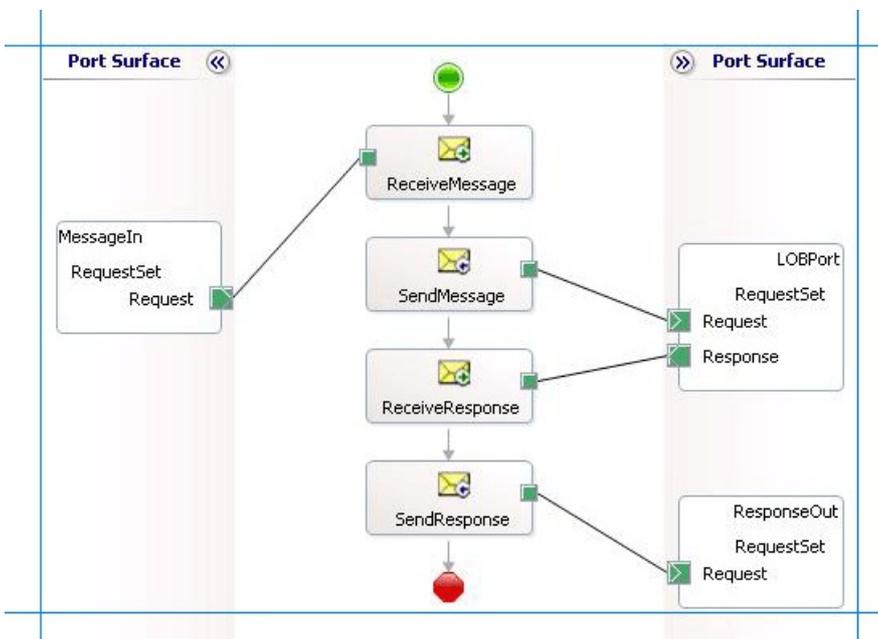
Use this	To do this
Response	<i>RequestSet.OracleEBSBindingRequestSets_FND.FNDRSSUB43Response</i>

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking request sets in Oracle E-Business Suite. In this orchestration, you will be dropping a request message at a defined receive location. The orchestration consumes this message and passes it on to Oracle E-Business Suite to invoke the **FNDRSSUB43** request set. The response for the request set is received from Oracle and is saved at another location. A typical orchestration to invoke a request set would contain:

- Send and Receive shapes to send messages to Oracle E-Business Suite and receive responses.
- A one-way receive port to receive request messages to send to Oracle E-Business Suite.
- A two-way send port to send request messages to Oracle E-Business Suite and receive responses.
- A one-way send port to send the responses from Oracle E-Business Suite to a folder.

A sample orchestration to invoke a request set resembles the following:



Adding Message Shapes

Specify the following properties for each of the message shapes. The names listed in the *Shape* column are the names of the message shapes as displayed in the preceding orchestration.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the *Port* column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values to be set to specify messages for action shapes and linking them to the ports. The names listed in the *Shape* column are the names of the message shapes as displayed in the previous orchestration.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.RequestSet.Request</i>

SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.RequestSet.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.RequestSet.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.RequestSet.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and then deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle E-Business Suite.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from Oracle E-Business Suite.
 - Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to Oracle E-Business Suite. You must also specify the action in the send port. For information about how to create send ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).

To invoke request sets using the Oracle E-Business adapter, you must set the right application context in which the operation is invoked. The Oracle E-Business adapter provides certain binding properties to specify the application context for any operation. You must set these binding properties on the WCF-Custom or WCF-OracleEBS port used for invoking request sets.

- If the **ClientCredentialType** binding property is set to **Database**, then you must specify the following binding properties to set the applications context.

Binding Property	Value

OracleUserName	Specify the name of an Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OracleUserName binding property when it connects to Oracle E-Business Suite. The user name is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the user name to be preserved or if you want to enter a user name containing special characters, you must specify the value within double quotes.
OraclePassword	The password for the Oracle E-Business Suite user. The Oracle E-Business adapter does not preserve the case of the value that you enter for the OraclePassword binding property when it connects to Oracle E-Business Suite. The password is passed to Oracle E-Business Suite using the standard rules of SQL*Plus. However, if you want the case of the password to be preserved or if you want to enter a password containing special characters, you must specify the value within double quotes.
OracleEBSResponsibilityName	The responsibility associated with the Oracle E-Business Suite user.

- If the **ClientCredentialType** binding property is set to **EBusiness**, you must have already specified Oracle E-Business credentials while establishing the connection. In such a case you must only specify value for the **OracleEBSResponsibilityName** binding property.

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For more information on how the adapter supports setting the application context, see [Setting Application Context](#).

Note

You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking the request sets. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to invoke the **FNDRSSUB43** request set is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message that conforms to the schema for invoking the **FNDRSSUB43** request set. For example, the request message to invoke the **FNDRSSUB43** request set is:

```
<FNDRSSUB43 xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSets/FND">
  <StartTime></ StartTime>
  <All_x0020_Requests_x0020_in_x0020_the_x0020_Set_STAGE10>
    <FNDMNAV xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetStage/FND/FNDRSSUB43">
      <Responsibility xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND">System Administrator</Responsibility>
      <ns3:Application xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND"></ns3:Application>
    </ns2:FNDMNAV>
    <ns2:FNDMMNNU xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetStage/FND/FNDRSSUB43">
      <ns3:Responsibility xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND">System Administrator</ns3:Responsibility>
      <ns3:Application xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND"></ns3:Application>
    </ns2:FNDMMNNU>
    <ns2:FNDMNFUN xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetStage/FND/FNDRSSUB43">
      <ns3:Responsibility xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND">System Administrator</ns3:Responsibility>
      <ns3:Application xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSetConcurrentProgram/FND/FNDRSSUB43/STAGE10/FND"></ns3:Application>
    </ns2:FNDMNFUN>
  </ns0:All_x0020_Requests_x0020_in_x0020_the_x0020_Set_STAGE10>
</ns0:FNDRSSUB43>
```

Note

The request message for invoking a request set requires some optional parameters such as SetRelClassOptions, SetPrintOptions, SetRepeatOptions, and SetNlsOptions. The request message provided here does not contain these optional parameters. For information on the complete request message, including optional parameters, see [Message Schemas for Request Sets](#).

The orchestration consumes the message, passes it on to the Oracle E-Business Suite, and receives a response. The response message is saved at the other file location specified as part of the orchestration. The response for the **FNDRSSUB43** request set resembles the following:

```
<?xml version="1.0" encoding="utf-8"?>
<FNDRSSUB43Response xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/RequestSets/FND">
  <FNDRSSUB43Result>2543208</FNDRSSUB43Result>
</FNDRSSUB43Response>
```

The response from the Oracle E-Business Suite contains a request ID. Request ID '0' denotes that the final submit operation on the request set failed. In such a case, you can specify other optional parameters in the request message, such as ContinueOnFail, to find out the reason of failure and to debug further.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server

The Oracle E-Business adapter exposes generic operations such as **ExecuteNonQuery**, **ExecuteReader**, and **ExecuteScalar**. You can use these operations to execute any SQL statement on Oracle database. These operations differ based on the kind of response you get for the SQL statement. For more information about how the adapter supports these operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

This topic demonstrates how to perform an **ExecuteReader** operation using the Oracle E-Business adapter with BizTalk Server. You can follow the same set of procedures described in this topic to perform **ExecuteNonQuery** and **ExecuteScalar** operations.

How to Invoke ExecuteReader operation on Oracle Database

Performing an operation on a Oracle database by using Oracle E-Business adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke the **ExecuteReader** operation on Oracle database, these tasks are:

1. Create a BizTalk project, and generate schema for the **ExecuteReader** operation.
2. Create messages in the BizTalk project for sending and receiving messages from Oracle database.
3. Create an orchestration to invoke the operation on Oracle database.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke the **ExecuteReader** operation on an Oracle database using the Oracle E-Business adapter with BizTalk Server. The **ExecuteReader** operation takes any SQL statement as a parameter and returns the result set for the operation as an array of DataSet. For this topic, we execute a SELECT statement on ACCOUNTACTIVITY table using the **ExecuteReader** operation. The ACCOUNTACTIVITY table is created by running the scripts provided with the samples. For more information about the script, see [Samples](#).

To demonstrate how to invoke **ExecuteReader** operation, schema is generated for the **ExecuteReader** operation. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. For more information about how to generate schemas, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>Execute_Reader.GenericOperation.ExecuteReader</i> , where <i>Execute_Reader</i> is the name of your BizTalk project. <i>GenericOperation</i> is the schema generated for the ExecuteReader operation.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

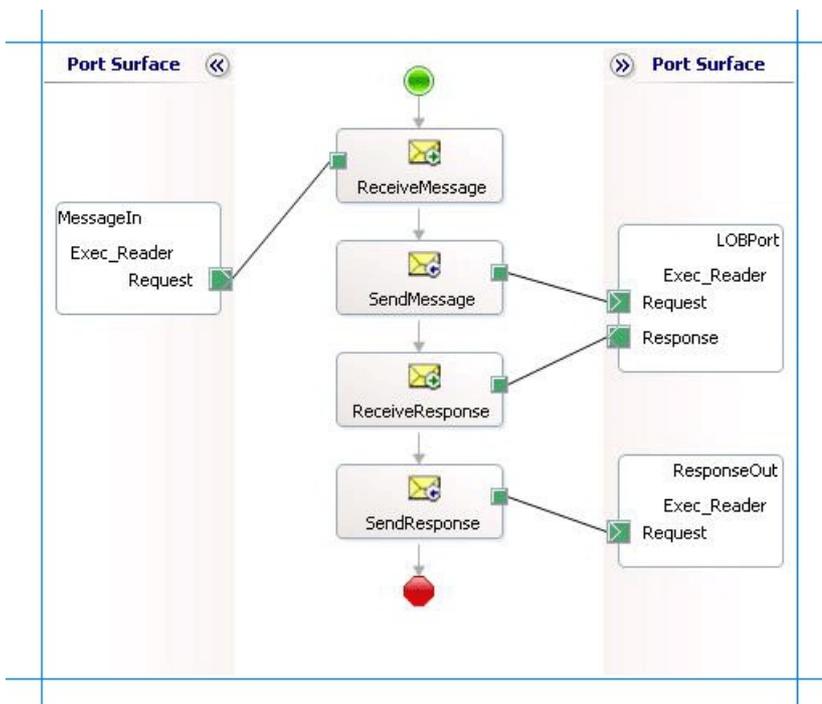
Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>Execute_Reader.GenericOperation.ExecuteReaderResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on Oracle database. In this orchestration, you drop a request message at a defined receive location. The Oracle E-Business adapter consumes this message and passes it on to Oracle database. The response from Oracle database is saved to another location. A typical orchestration for invoking generic operations such as **ExecuteReader** would contain:

- Send and Receive shapes to send and receive messages from an Oracle database.
- A two-way receive port to send and receive messages from an Oracle database.
- A one-way send port to send response from Oracle database to a folder.

A sample orchestration for invoking an **ExecuteReader** operation resembles the following:



Adding Message Shapes

You need to configure the following properties for each of the message shapes. The names listed in the Shape column correspond to the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

For each of the logical ports, configure the properties in the table below. The names listed in the Port column correspond to the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect Them to Ports

The following table specifies the property values to specify messages for action shapes and to link the messages to the ports. The names listed in the **Shape** column correspond to the names of the message shapes as displayed in the orchestration diagram earlier.

After you have configured these properties, the message shapes and ports are connected, and your orchestration is complete.

Then, you must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.Exec_Reader.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Exec_Reader.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Exec_Reader.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.Exec_Reader.Request</i>

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Oracle database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Oracle database.
 - Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to the Oracle database. You must also specify the action in the send port. For information about how to create send ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).

◆ Important

As part of generic operations, if you are executing operations on objects, for example stored procedures, functions, interface tables, or interface views, which belong to an Oracle E-Business Suite application, you must set the application context by specifying the requisite binding properties. For more information about setting the application context, see [Setting Application Context](#).

You can set the application context either by specifying the binding properties or by setting the message context properties exposed by the Oracle E-Business adapter. For instructions on how to set the binding properties, see [Specifying Binding Properties](#). For instructions on how to set the application context using message context properties, see [Set the Application Context Using Message Context Properties](#).

📌 Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application before invoking **ExecuteReader** operation on Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-OracleEBS send port to send messages to Oracle database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the **ExecuteReader** operation you generated earlier. For example, the request message to execute a SELECT statement using an **ExecuteReader** operation is:

```
<ExecuteReader xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/">
  <Query>SELECT * FROM ACCOUNTACTIVITY</Query>
</ExecuteReader>
```

See Message Schemas for ExecuteReader, ExecuteScalar, and ExecuteNonQuery Operations for more information about the request message schema for invoking an **ExecuteReader** operation using the Oracle E-Business adapter.

The orchestration consumes the message and sends it to the Oracle database. The response from the Oracle database is saved at the other FILE location defined as part of the orchestration. The response for the **ExecuteReader** operation contains a result set as a DataSet. For example, the response from the Oracle database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<ExecuteReaderResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/">
  <ExecuteReaderResult>
    <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
      <xs:element msdata:IsDataSet="true" name="NewDataSet">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                  <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="AMOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="DESCRIPTION" type="xs:string" />
                  <xs:element minOccurs="0" name="TRANSDATE" type="xs:dateTime" />
                  <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
```

```
<NewDataSet xmlns="">
  <NewTable>
    <TID>1</TID>
    <ACCOUNT>100001</ACCOUNT>
    <AMOUNT>500</AMOUNT>
    <DESCRIPTION />
    <TRANSDATE>2008-08-04T13:04:20</TRANSDATE>
    <PROCESSED>n</PROCESSED>
  </NewTable>
  <NewTable>
    .....
    .....
  </NewTable>
  .....
  .....
</NewDataSet>
</diffgr:diffgram>
</ExecuteReaderResult>
</ExecuteReaderResponse>
```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. After you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Polling Oracle E-Business Suite Using BizTalk Server

You can configure the Oracle E-Business adapter to receive polling-based messages from Oracle database. The adapter provides two ways of polling the Oracle database:

- **Using SELECT statements.** You can specify a simple SELECT statement to poll the Oracle database. The adapter executes the SELECT statement at specified intervals and returns the result to the adapter clients.
- **Using stored procedures.** You can specify a stored procedure to poll the Oracle database. The adapter executes the stored procedure at specified intervals and returns the result to the adapter clients.

The key difference in the two approaches is the way adapter clients specify a polling statement that the adapter uses to poll the Oracle database. While the polling statement for the first approach is a simple SELECT statement, the polling statement for the stored procedure approach is a request message that executes the stored procedure. Adapter clients specify the polling statement, for either approach, for the **PollingInput** binding property. For more information about the binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

The topics in this section provide instructions on how to poll using a SELECT statement and a stored procedure.

In This Section

- [Polling Using SELECT Statement](#)
- [Polling Using Stored Procedures](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Polling Using SELECT Statement

You can configure the Oracle E-Business adapter to receive periodic data-change messages by using a SELECT statement to continuously poll the interface tables, interface views, tables and views in Oracle E-Business Suite. You can specify a SELECT statement as a polling statement that the adapter executes periodically to poll Oracle E-Business Suite. You can also specify a post-poll PL/SQL code block that the adapter executes after the polling statement is executed.

To enable polling, you must specify certain binding properties on the WCF-Custom or WCF-OracleEBS receive port. For more information about how the adapter supports polling, see [Support for Inbound Calls Using Polling](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling Operations](#).

Configuring a Polling Operation with Oracle E-Business Suite Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure the adapter to receive data change messages. You must specify these binding properties while configuring the receive port in BizTalk Server Administration console.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the SELECT statement you specify for the PollingInput binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle E-Business adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingInput	Specifies the polling statement. To poll using a SELECT statement, you must specify a SELECT statement for this binding property. The default is null. You must specify a value for PollingInput binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed.
PollWhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to poll the Oracle database, read further.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving data change messages using SELECT statements, create a BizTalk project and generate schema for the **Poll** operation for table you want to poll. In this topic, we generate schema for the **Poll** operation for the **MS_SAMPLE_EMPLOYEE** interface table in the **Application Object Library**

application. This table is created when you run the `create_apps_artifacts.sql` script provided with the samples to create these objects in Oracle E-Business Suite.

Next, we will use content-based routing (CBR) in BizTalk Server to configure an application with a receive port that will receiving polling messages from the **MS_SAMPLE_EMPLOYEE** interface table, and then route it to a send port. In this case, we will create a filter on the send port that checks the receive location specified, and the message is routed to the send port.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the interface table being polled (`MS_SAMPLE_EMPLOYEE`) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE
```

This ensures that the adapter executes the polling statement only when the `MS_SAMPLE_EMPLOYEE` interface table has some records.

- Specify a SELECT statement for the **PollingInput** binding property. This statement retrieves all the rows in the `MS_SAMPLE_EMPLOYEE` interface table. In this example, you can set this binding property as:

```
SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE
```

Note

For information about the FOR UPDATE clause used in the SELECT statement, see [Specifying a FOR UPDATE Clause in the Polling Statement](#).

- Specify a DELETE statement as part of the **PostPollStatement** binding property. This statement will delete all data from `MS_SAMPLE_EMPLOYEE` interface table. In this example, you can set this binding property as:

```
DELETE FROM MS_SAMPLE_EMPLOYEE
```

Once this happens, the next time the statement specified for **PollingInput** will be executed, it will not fetch any data.

- Until more data is added to the `MS_SAMPLE_EMPLOYEE` interface table, you will not get any polling messages. So, you must repopulate the `MS_SAMPLE_EMPLOYEE` interface table with new records. You can do so by running the `insert_apps_data.sql` script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

How to Receive Data-change Messages from Oracle E-Business Suite

Performing an operation on Oracle database using Oracle E-Business adapter with BizTalk Server involves the following procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to poll Oracle E-Business Suite using a SELECT statement, these tasks are:

1. Create a BizTalk project, and generate schema for the **Poll** operation for the interface table you want to poll.
2. Build and deploy the BizTalk project.
3. Configure the BizTalk application by creating receive and send ports. Additionally, add filter on the send port so that it checks the receive location specified in the receive port, and the polling message is routed to the send port.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

4. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `PollingUsingSelectStatement`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

You must generate the schema for the **Poll** operation on the `MS_SAMPLE_EMPLOYEE` interface table in the **Application Object Library** application. Perform the following tasks while generating the schema using the Consume Adapter Service Add-in.

- Select the contract type as **Service (Inbound operation)**.
- Generate schema for the **Poll** operation on the `MS_SAMPLE_EMPLOYEE` interface table. You can select the operation and the interface table either from the **Application-Based View** node or the **Artifact -Based View** node.

For more information about how to generate schema, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

Building and Deploying the BizTalk Project

You must now build the BizTalk solution, and deploy it to a BizTalk Server. For information about deploying the solution to BizTalk Server, see <http://go.microsoft.com/fwlink/?LinkId=107182>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the application is listed under the **Applications** node in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. As part of configuring the application, you must create a receive port and a send port in the application, and then add filter to the send port so that all the messages from the receive port are routed to the send port.

Configuring an application involves:

- Selecting a host for the application.
- Creating receive and send ports.

Creating a Receive Port

You must create a WCF-Custom or WCF-OracleEBS one-way receive port, which polls Oracle using the SELECT statement specified for the **PollingInput** binding property. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). While creating the receive port, ensure you specify the following binding properties.

For Polling

Binding Property	Value
InboundOperationType	Set this to Polling .
PolledDataAvailableStatement	For this example, set this binding property to: <pre>SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE</pre>
PollingAction	Retrieve the polling action from the schema generated for the Poll operation on <code>MS_SAMPLE_EMPLOYEE</code> interface table. For this example, set this binding property to InterfaceTables/Poll/FND/APPS/MS_SAMPLE_EMPLOYEE .

PollingInput	<p>For this binding property, specify a SELECT statement to retrieve all records from the MS_SAMPLE_EMPLOYEE interface table. For this example, set this binding property to:</p> <pre data-bbox="280 185 1501 264">SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE</pre>
PostPollState ment	<p>Specify the post-poll statement to delete all data from the MS_SAMPLE_EMPLOYEE interface table. For this example, set this binding property to:</p> <pre data-bbox="280 398 1501 477">DELETE FROM MS_SAMPLE_EMPLOYEE</pre>

For Setting Application Context

If you are performing operation on Oracle E-Business Suite artifacts, you must specify values for the appropriate binding properties to set the application context. For more information about application context and the binding properties required for setting application context, see [Setting Application Context](#).

For this example, specify appropriate values for the **oracleUserName**, **oraclePassword**, and **oracleEBSResponsibility** binding properties.

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle E-Business adapter. You can do so by adding the service behavior while configuring the receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

Creating a Send Port

Define a location on the hard disk and create a corresponding FILE send port where BizTalk Server will drop the messages from Oracle. These messages will be in response to the polling statement that you specify for the receive port. For information about how to create send ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#).

You must also add a filter on the send port to route messages from the receive location. To add filter to the send port:

1. Double-click the send port to open the **Send Port Properties** dialog box.
2. In the **Send Port Properties** dialog box, click the **Filters** tab.
3. Specify the following values:
 - In the **Property** list, click **BTS.ReceivePortName**.
 - In the **Operator** list, click **=**.
 - In the **Value** field, specify the receive port name.
4. In the **Send Port Properties** dialog box, click **OK**.

Starting the Application

You must start the BizTalk application for polling Oracle E-Business Suite. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleEBS one-way receive port, which polls Oracle using the SELECT statement specified for the **PollingInput** binding property, is running.
- The FILE send port, which receives messages from Oracle database, is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** which returns a positive value indicating the adapter to execute the statement specified for **PollingInput** binding property.
- The adapter executes the SELECT statement for the **PollingInput** binding property and returns all the rows in the MS_SAMPLE_EMPLOYEE interface table. The response from Oracle E-Business Suite resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Poll xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/FND/APPS/M
S_SAMPLE_EMPLOYEE">
  <DATA>
    <SelectRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecor
d/APPS/MS_SAMPLE_EMPLOYEE">
      <EMP_NO>10002</EMP_NO>
      <NAME>JEFF PRICE</NAME>
      <DESIGNATION>MANAGER</DESIGNATION>
      <SALARY>25000</SALARY>
      <JOIN_DATE>2007-12-15T00:00:00</JOIN_DATE>
    </SelectRecord>
    <SelectRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecor
d/APPS/MS_SAMPLE_EMPLOYEE">
      <EMP_NO>10003</EMP_NO>
      <NAME>DON HALL</NAME>
      <DESIGNATION>ACCOUNTANT</DESIGNATION>
      <SALARY>12000</SALARY>
      <JOIN_DATE>2005-10-29T00:00:00</JOIN_DATE>
    </SelectRecord>
    ...
    <SelectRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecor
d/APPS/MS_SAMPLE_EMPLOYEE">
      ...
    </SelectRecord>
    ...
  </DATA>
</Poll>
```

- The adapter executes the post-poll statement, which deletes all the data from MS_SAMPLE_EMPLOYEE interface table.
- After the polling interval, the adapter again executes **PolledDataAvailableStatement**. Because the MS_SAMPLE_EMPLOYEE interface table has no records now, **PolledDataAvailableStatement** does not return a positive value and hence the adapter does not execute the statement specified for the **PollingInput** binding property. As a result, adapter client does not get any polling message.
- The adapter client will not get any more polling messages until some records are explicitly inserted into the MS_SAMPLE_EMPLOYEE interface table. To insert more records, you can run the insert_apps_data.sql script provided with the samples. After you run this script, the next time **PolledDataAvailableStatement** is executed, it returns a positive value. As a result, the adapter executes the polling statement and adapter clients again receive a polling message.

Note

The Oracle E-Business adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports and receive ports. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling Oracle E-Business Suite Using BizTalk Server](#)

Polling Using Stored Procedures

You can configure the Oracle E-Business adapter to receive periodic data-change messages by using stored procedures to continuously poll the Oracle database. You can specify a stored procedure as a polling statement that the adapter executes periodically to poll the Oracle database.

To enable polling, you must specify certain binding properties on the WCF-Custom or WCF-OracleEBS receive port. For more information about how the adapter supports polling, see [Support for Inbound Calls Using Polling](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling Operations](#).

Configuring a Polling Operation with Oracle E-Business Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure the adapter to receive data-change messages. You must specify these binding properties while configuring the WCF-Custom or WCF-OracleEBS receive port in the BizTalk Server Administration console.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform a Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the stored procedure you specified for the PollingInput binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle E-Business adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingInput	Specifies the polling statement. To poll using a stored procedure, you must specify the entire request message for this binding property. The request message must be the same that you send to the adapter for invoking the stored procedure as an outbound operation. The default is null. You must specify a value for PollingInput binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed.
PollWhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to poll the Oracle database, read the following sections.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving data change messages using stored procedures, create a BizTalk project and generate schema for the stored procedure you want to use to poll the Oracle database. In this topic, we use the GET_ACTIVITYS stored procedure to poll the ACCOUNTACTIVITY table. This stored procedure is

available with the ACCOUNT_PKG package. You can run the SQL scripts provided with the samples to create these objects in the database.

Note

The orchestration in this topic polls the ACCOUNTACTIVITY table, which is a base database table created by running the scripts provided with the samples. You must perform similar procedures as described in this topic to poll any other table, including interface tables.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the table being polled (ACCOUNTACTIVITY) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM ACCOUNTACTIVITY
```

This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.

- Execute a stored procedure, GET_ACTIVITYS, by providing the request message as part of the **PollingInput** binding property. This stored procedure will retrieve all the rows in the ACCOUNTACTIVITY table and you will get a response message from the adapter.
- EXECUTE a PL/SQL block as part of the **PostPollStatement** binding property. This statement will move all data from ACCOUNTACTIVITY table to another table in the database. Once this happens, the next time **PollingInput** will be executed, it will not fetch any data and hence the GET_ACTIVITYS stored procedure will return an empty response message.
- Until more data is added to the ACCOUNTACTIVITY table, you will continue to get empty response messages. So, you must repopulate the ACCOUNTACTIVITY table with new records. You can do so by running the more_activity_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

How to Receive Data-change Messages from Oracle

Performing an operation on Oracle database using Oracle E-Business adapter with BizTalk Server involves the following procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to poll Oracle database using a stored procedure, these tasks are:

1. Create a BizTalk project, and generate schema for the stored procedure you want to use for polling.
2. Create a message in the BizTalk project for receiving messages from Oracle database.
3. Create an orchestration to receive messages from Oracle database and save them to a folder.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, `PollingUsingStoredProc`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

You must generate the schema for the `GET_ACTIVITYS` operation. Perform the following tasks while generating the schema using the Consume Adapter Service Add-in.

- Select the contract type as **Service (Inbound operation)**.
- Generate schema for the **GET_ACTIVITYS** procedure.

For more information about how to generate schema, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from Oracle.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

U s e t h is	To do this
Id e nt ifi er	Type Receive .

From the drop-down list, expand **Schemas**, and select *Polling.OracleEBSBindingSchema*, where *Polling* is the name of your BizTalk project. *OracleEBSBindingSchema* is the response schema generated for the **GET_ACTIVITYS** stored procedure.

Important

Because polling is a one way operation, the schema generated by the adapter does not contain a response node, and hence there is only one root node in the schema. If you use such schemas for a message type, you must identify the schema by the filename of the generated schema. For example, if you create schema for a two-way operation, the nodes in the schema file with a name *OracleEBSBindingSchema* may look like "Request" and "Response". If you want to create a message in the orchestration that maps to the request schema, you can identify the schema in the list by looking for *OracleEBSBindingSchema.Request*. However, in the case of polling operation, because the only node is "Poll", it is not easy to identify the schema you want to map to because schemas with single nodes are not listed as <schemafilename>.<rootnodename>. Instead, such schemas are listed by only the filename. In such a case, the only way to identify the schema is by the schema filename, for example, *OracleEBSBindingSchema*.

The Consume Adapter Service Add-in generates schema for both inbound and outbound operations for the GET_ACTIVITYS stored procedure. You must use the schema for inbound operation to:

- Map the message created as part of the orchestration.
- To retrieve the action you must specify for the **PollingAction** binding property at run-time.

You must use the schema for the outbound operation to get the request message you must specify as part of the **PollingInput** binding property.

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from Oracle. In this orchestration, the adapter receives the response by executing the stored procedure for which you specified the request message as part of the **PollingInput** binding property. The response message for the stored procedure is saved to a FILE location. A typical orchestration for polling Oracle database would contain:

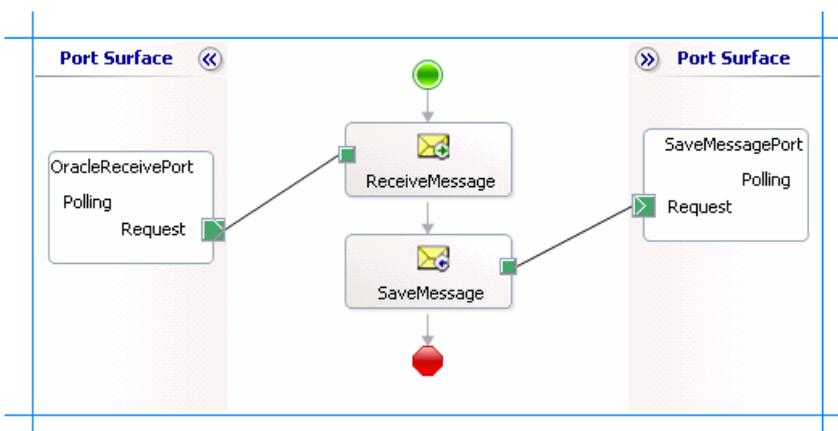
- Receive and Send shapes to receive messages from Oracle and send to a FILE port, respectively.
- A one-way receive port to receive messages from Oracle database.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from Oracle database.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the

names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SaveMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveMessage</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleReceivePort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleReceivePort</i> • Set Type to <i>OracleReceivePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>OracleReceivePort.Polling.Request</i>
SaveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SaveMessagePort.Polling.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at

<http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding FILE port where the BizTalk orchestration will drop the messages from Oracle. These messages will be in response to the polling statement that you specify for the receive port.
 - Define a physical WCF-Custom or WCF-OracleEBS one-way receive port. This port polls the Oracle database. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). Make sure you specify the following binding properties for the receive port.

Binding Property	Value
Inbound OperationType	Set this to Polling .
PolledDataAvailableStatement	<p>For this example, set this binding property to:</p> <pre>SELECT COUNT (*) FROM ACCOUNTACTIVITY</pre> <p>This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.</p>
Polling Action	Retrieve the polling action from the schema generated for the inbound message for the GET_ACTIVITY_S procedure. For this example, set this binding property to PollingPackageApis/APPS/ACCOUNT_PKG/GET_ACTIVITIES .
PollingInput	<p>For this binding property, specify the request message to invoke the GET_ACTIVITY_S stored procedure. You can get the request message from the schema for the outbound operation generated by the Consume Adapter Service Add-in. You must provide the entire XML message as an input for this binding property. For this example, set this binding property to:</p> <pre><GET_ACTIVITYS xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/APPS/ACCOUNT_PKG"> <INRECS>OPEN ? FOR SELECT * FROM ACCOUNTACTIVITY</INRECS> </GET_ACTIVITYS></pre> <p>The GET_ACTIVITY_S stored procedure takes an input REF CURSOR as parameter.</p>
PostPollStatement	<p>Specify the post-poll statement to move all data from ACCOUNTACTIVITY table to another table. For this example, set this binding property to:</p> <pre>BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;</pre>

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

◆ Important

If you are polling an interface table, you must set the application context by specifying the requisite binding properties. For more information about setting the application context see [Setting Application Context](#).

📌 Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle E-Business adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-OracleEBS receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

Starting the Application

You must start the BizTalk application for polling Oracle database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleEBS one-way receive port, which polls Oracle using the stored procedure specified for the **PollingInput** binding property, is running.
- The FILE send port, which receives messages from Oracle database, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** which returns a positive value indicating the adapter to execute the statement specified for **PollingInput** binding property.
- The adapter executes the GET_ACTIVITYS stored procedure specified for the **PollingInput** binding property and returns all the rows in the ACCOUNTACTIVITY table. The response from Oracle database resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<GET_ACTIVITYS xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PollingPackageApis/APPS/ACCOUNT_PKG">
  <OUTRECS>
    <OUTRECSRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ReferencedRecordTypes/APPS/ACCOUNT_PKG/GET_ACTIVITYS/APPS/GET_ACTIVITYS">
      <TID>1</TID>
      <ACCOUNT>100001</ACCOUNT>
      <AMOUNT>500</AMOUNT>
      <DESCRIPTION />
      <TRANSDATE>2008-06-21T15:52:19</TRANSDATE>
      <PROCESSED>n</PROCESSED>
    </OUTRECSRecord>
    <OUTRECSRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ReferencedRecordTypes/APPS/ACCOUNT_PKG/GET_ACTIVITYS/APPS/GET_ACTIVITYS">
      .....
      .....
    </OUTRECSRecord>
    .....
    .....
  </OUTRECS>
</GET_ACTIVITYS>
```

- The adapter executes the post-poll statement, which moves all the data from ACCOUNTACTIVITY table to another table.
- After the polling interval, the adapter again executes **PolledDataAvailableStatement**. Because ACCOUNTACTIVITY table has no records now, **PolledDataAvailableStatement** does not return a positive value and hence the adapter does not execute the statement specified for the **PollingInput** binding property. As a result, adapter client does not get any polling message.
- The adapter client will not get any more polling messages until some records are explicitly inserted into the ACCOUNTACTIVITY table. To insert more records, you can run the more_activity_data.sql script provided with the samples. After you run this script, the next time **PolledDataAvailableStatement** is executed, it returns a positive value. As a result, the adapter executes the polling statement and adapter clients again receive a polling message.

Note

The Oracle E-Business adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling Oracle E-Business Suite Using BizTalk Server](#)

Receiving Database Change Notifications Using BizTalk Server

You can configure the Oracle E-Business adapter to receive database change notification messages from Oracle E-Business Suite. You can specify a SELECT statement that the adapter uses to register for notifications with Oracle E-Business Suite. The adapter receives a notification message when the result set for the SELECT statement, registered for notification, changes. For more information about how the adapter supports notification, see [Receiving Database Change Notifications](#).

Following are some scenarios in which you can configure the Oracle E-Business adapter with BizTalk Server to receive notifications from Oracle E-Business Suite:

- Adapter clients get only “incremental” notification, for example, only for those changes that were made to a database table since the last notification.
- If a large number of rows are inserted into a database table, the adapter clients can configure multiple receive locations to load-balance receiving notifications.

After the adapter clients receive a notification message, they can perform specific tasks based on the kind of notification received. For example, a BizTalk orchestration can be designed in such a way that it performs one set of tasks if an insert notification is received and another set of tasks if an update notification is received.

Caution

If there is a network outage between the Oracle database and the adapter client, the notifications will not be sent to the adapter clients for the changes done on the Oracle database during the period of network outage, and thereafter. Therefore, you must use the Polling operation instead of the Notification operation for critical scenarios.

The topics in this section provide information on how to configure the adapter for each of these scenarios. To start getting notifications from Oracle E-Business Suite using the Oracle E-Business adapter, you must specify certain binding properties. For more information about the binding properties related to notifications, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For more information about structure of notification messages, see [Message Schemas for the Notification Operation](#).

For receiving notifications from Oracle E-Business Suite, make sure:

- You use the adapter to connect to Oracle database version 10.2 or later. Oracle database versions prior to 10.2 do not support notifications.
- The **change notification** privilege is required for receiving database change notifications. To configure this privilege, connect to Oracle database using administrative privileges and then type the following command on the SQL prompt.

```
grant change notification to <user name>
```

- Decide on a TCP port you want ODP.NET to use for receiving database change notifications from Oracle database. Add that port to Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkID=196959>. You must provide the same port number for the **NotificationPort** binding property. For more information about the binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

In This Section

- [Considerations for Receiving Database Change Notifications Using the Adapter](#)
- [Processing Notification Messages to Perform Specific Tasks](#)
- [Receiving Database Change Notifications Incrementally Using BizTalk Server](#)
- [Receiving Database Change Notifications On Multiple Receive Locations](#)

- [Receiving Database Change Notifications After a Receive Location Breakdown](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Considerations for Receiving Database Change Notifications Using the Adapter

This topic provides some considerations and best practices that you must keep in mind while using the Oracle E-Business adapter to receive database notifications from an Oracle database.

Considerations While Using the Adapter to Receive Notifications

You must consider the following while using the Oracle E-Business adapter to receive query notifications.

- The Oracle E-Business adapter simply passes on the notification, which it receives from the Oracle database, to the adapter clients. The adapter does not distinguish between the notifications for different operations, i.e., the adapter does not have any information whether a particular notification is for an Insert operation or an Update operation.
- The notification message for an operation is not affected by the number of records affected by that operation. For example, irrespective of the number of records inserted in an Oracle database table, the adapter clients receive only one notification message.
- We recommend that the adapter client application contain the logic to interpret the kind of notification received from the Oracle database. The adapter client applications can do so by extracting the information in the **<Info>** element of the received notification message. Here's an example of a notification message received for an Insert operation.

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>1</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Insert</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

Notice the value within the **<Info>** element. This value provides information on the operation for which the notification message was received. Your application should have the functionality to extract the value within the **<Info>** element and then based on the value, perform subsequent tasks. The topic [Processing Notification Messages to Perform Specific Tasks](#) has instructions on how to extract the value within the **<Info>** element.

- Ideally, after the client application receives a notification, it should update the record for which the notification is already received so that the subsequent notifications are not for the same record. For example, consider an **ACCOUNTACTIVITY** table that has a **Processed** column. For all new records inserted into the **ACCOUNTACTIVITY** table, the value in the **Processed** column is always 'n'. For example, after an insert operation, the records in the **ACCOUNTACTIVITY** table will look like the following:

Account Transaction ID	Processed
10001	n

To get notifications for the newly inserted record, the adapter client will set the **NotificationStatement** binding property as:

```
SELECT * FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'
```

After, receiving the notification, the client application must set the value of the **Processed** column to 'y' so that the notification statement does not operate on the record that was already notified for. So, to achieve this, the client application must perform an Update operation on the **ACCOUNTACTIVITY** table. After the Update operation, the same record in the **ACCOUNTACTIVITY** table will look like the following:

Account Transaction ID	Processed
10001	y

Interestingly, the Update operation will again send a notification to the adapter client and the whole process will be repeated again. So, the client application must have the required logic to discard such unwanted notifications.

- If the **NotifyOnListenerStart** binding property is true, the adapter will send a notification to the adapter client every time the receive location starts. For more information on how to use the binding property and interpret the notification message, see [Receiving Database Change Notifications After a Receive Location Breakdown](#).

Typical Orchestration for Receiving Notifications

This section outlines the typical orchestration flow for receiving notifications using the Oracle E-Business adapter.

1. The first thing that the orchestration must do is to check the kind of notification received. The things to check for are:
 - Whether the notification was received for the receive location restart.
 - Whether the notification was received for an operation on a database table, such as Insert, Update, or Delete.

The orchestration must include an **Expression** shape, and within that an xpath query, to decide what kind of message is received.

2. After the notification type is available, the orchestration must include a decision block to perform specific actions based on the type of notification received. To achieve this, the orchestration must include a **Decide** shape. The **Decide** shape consists of a **Rule** block and an **Else** block. Within the **Rule** block, you must specify the condition and then include orchestration shapes to perform certain operations if the condition is met. Within the **Else** block, you must include orchestration shapes to perform certain operations if the condition is *not* met.

The preceding recommendations are described in detail in [Processing Notification Messages to Perform Specific Tasks](#).

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Processing Notification Messages to Perform Specific Tasks

You can use the Oracle E-Business adapter to receive notifications for changes to the Oracle database tables. However, the adapter only sends you a notification that some records were inserted, updated, or deleted in a certain database table. Any post-processing on those records must be handled by the client applications themselves. This topic presents a scenario-based description on how to process the records in the table based on the kind of notification received from the Oracle database.

Scenarios for Performing Subsequent Actions After Receiving Notification

Following are a couple of scenarios in which the adapter clients must perform certain post-notification tasks.

- **Scenario 1.** Consider a scenario where the adapter client must perform certain tasks based on the kind of notification you receive from the Oracle database. For example, the client application must update the records in table "A" if records are inserted in table "B". Similarly, the client application must delete records from table "A" if records are deleted from table "B".

In this scenario, from the notification message received, the adapter clients must extract the type of notification to decide whether the notification was for an insert operation or a delete operation. Once the notification type is ascertained, the adapter clients must perform subsequent actions to insert or update the relevant tables.

- **Scenario 2.** Consider a scenario where the receive location that receives notification messages for changes to a table goes down. While the receive location is down, some records are added to the table. However, for these records the adapter client does not receive any notification. When the receive location is back up, the adapter notifies the client by sending a specific message, and then the client application must look for all the records that were inserted in the database table while the receive location was down.

In this scenario, from the notification message received, the adapter clients must extract the information regarding whether the notification is for a change to a database table or for the receive location starting. If the notification is for the receive location starting, the adapter clients must implement the logic to process the records that might have been inserted, updated, or deleted while the receive location was down.

Note

These are just some example scenarios that are listed for a better understanding of how to use the notification feature in the Oracle E-Business adapter. However, the basic set of tasks required to extract the type of notification received will be similar for all scenarios. This topic provides instructions on how to extract the type of notification from a notification message.

How This Topic Demonstrates Receiving Notification Messages

In this topic, to demonstrate how to process notification messages to perform subsequent tasks, we consider a basic scenario where an adapter client uses BizTalk application to receive notification messages for changes to the ACCOUNTACTIVITY table. After the notification is received, the client filters the type of notification received and performs subsequent action. To demonstrate a very basic scenario, let us consider that the adapter client copies the notification messages to different folders based on the kind of notification received. Therefore:

- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.
- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotificaion folder.

To achieve this as part of a BizTalk application, the orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the information about the kind of notification message received.

- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.

Configuring Notifications with the Oracle E-Business Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure receiving notifications from Oracle E-Business Suite. You must specify these binding properties when configuring the receive port in the BizTalk Server Administration console.

Note	
You may choose to specify these binding properties when generating the schema for the Notification operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-OracleEBS receive port with the binding properties already set. For more information about creating a WCF-custom or WCF-OracleEBS port using the binding file, see Configuring a Physical Port Binding Using a Port Binding File .	

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the SELECT statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified SELECT statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to receive notifications from Oracle E-Business Suite, read further.

How to Receive Notification Messages from Oracle E-Business Suite

Performing an operation on Oracle E-Business Suite using the Oracle E-Business adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** inbound operation. Optionally, you can specify values for the **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from Oracle E-Business Suite.
3. Create an orchestration as described in the preceding section.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Note	
For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-OracleEBS receive port. Two-way receive ports are not supported for inbound operations.	

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Notification** inbound operation. For more information about how to generate the schema, see [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#).

Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.

Defining Messages and Message Types

The schema that you generated in the previous section describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. After the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive notifications from the Oracle database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>Process_Notification.OracleEBSBinding.Notification</i> , where <i>Process_Notification</i> is the name of your BizTalk project. <i>OracleEBSBinding</i> is the schema generated for the Notification operation.

Setting up the Orchestration

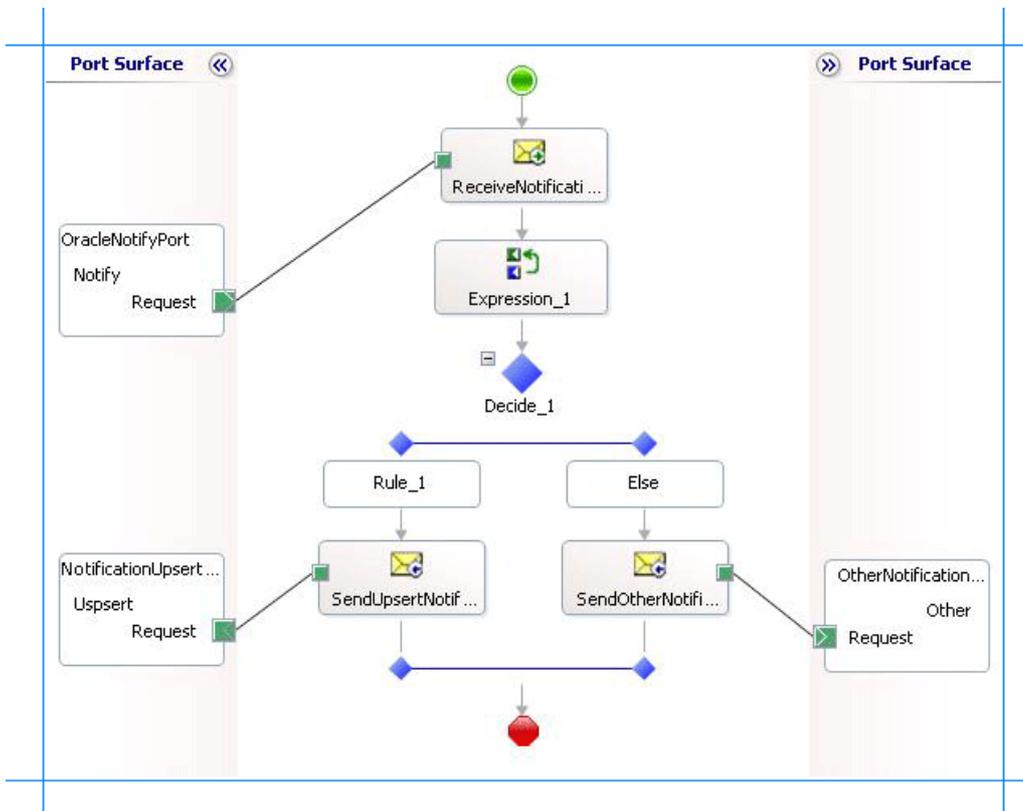
You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from the Oracle database and then performing tasks based on the type of notification received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The xpath query specified within the Expression shape extracts the type of notification into a variable, say **NotificationType**. The Decide shape uses the value in this variable to decide on the kind of notification received and takes the appropriate "path" to perform subsequent operations. As mentioned in the preceding section, the orchestration will perform the following operations based on the kind of notification message received.

- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.
- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotificaiton folder.

So, your orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the kind of notification received.
- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.
- Receive shape.

A sample orchestration resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the **Shape** column are the names of the message shapes as displayed in the orchestration diagram.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveNotification</i> • Set Activate to <i>True</i>

Adding an Expression Shape

The purpose of including an Expression shape in the orchestration is to have an xpath query to extract the kind of notification message received. Before creating an xpath query, let us look at the format of a notification message. A typical notification message resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>1</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
  <Info>Insert</Info>
  <ResourceNames>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.ACCOUNT
ACTIVITY</string>
  </ResourceNames>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

As you see, the information about the type of the notification is available within the `<Info>` tag, within the parent `<Notification>` tag. So, as part of this expression shape you must:

- Create a variable that contains the value within the `<Info>` tag and set its type to `System.String`. For more information about creating variables, see <http://go.microsoft.com/fwlink/?LinkId=102357>.

For this topic, name the variable as **NotificationType**.

- Create an xpath query to extract the value from the `<Info>` tag. The xpath query will resemble the following:

```
NotificationType = xpath(NotifyReceive, "string(/*[local-name()='Notification']/*[local
-name()='Info']/text())");
```

In this xpath query, **NotifyReceive** is the message you created for receiving notification messages. The excerpt within the `string` function indicates that the query must extract the value within the `<Info>` tag, which in turn is within the `<Notification>` tag. Finally, the value extracted by the query is assigned to the **NotificationType** variable.

Adding a Decide Shape

The purpose of adding a Decide shape is to include a decision block in the orchestration to decide what subsequent operations to perform based on the kind of notification message received. The decision is made on the basis of the value of the **NotificationType** variable. In this topic, the orchestration makes a decision based on the kind of notification message received. So, the condition in the Rule shape is specified as follows:

```
NotificationType.Equals("Insert") | NotificationType.Equals("Update")
```

This condition suggests that if the value for **NotificationType** variable is Insert or Update, the orchestration will perform one set of tasks. If the value of **NotificationType** variable is anything else, the orchestration will perform other set of tasks.

As mentioned in the preceding sections, to demonstrate a simple approach, the orchestration will copy messages to different folders based on the notification message type. So, within the Rule and Else blocks, you must add Send shapes to send the messages to different ports. For this topic, name the Send shape in the Rule block as **SendUpsertNotification** and the Send shape in the Else block as **SendOtherNotification**.

Adding Ports

You must now add the following logical ports to the orchestration:

- One-way receive port to receive notification messages from the Oracle database.
- One-way send port to send notification messages for Insert and Update operations to a specific folder.
- One-way send port to send notification messages for any other operations to a specific folder.

Make sure you specify the following properties for each of the logical ports. The names listed in the **Port** column are the names of the ports as displayed in the orchestration diagram shown earlier in this topic.

Port	Properties
OracleNotifyPort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleNotifyPort</i> • Set Type to <i>OracleNotifyPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
NotificationUpsertPort	<ul style="list-style-type: none"> • Set Identifier to <i>NotificationUpsertPort</i> • Set Type to <i>NotificationUpsertPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>
OtherNotificationPort	<ul style="list-style-type: none"> • Set Identifier to <i>OtherNotificationPort</i> • Set Type to <i>OtherNotificationPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the **Shape** column are the names of the message shapes as displayed in the orchestration diagram shown earlier in this topic.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>OracleNotifyPort.Notify.Request</i>
SendUpsertNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>NotificationUpsertPort.Upsert.Request</i>
SendOtherNotification	<ul style="list-style-type: none"> • Set Message to <i>Select</i> • Set Operation to <i>OtherNotificationPort.Other.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-OracleEBS one-way receive port. This port listens for notifications coming from Oracle E-Business Suite. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). Make sure you specify the following binding properties for the receive port.

◆Important
 You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-OracleEBS receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>◆Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre style="border: 1px solid black; padding: 5px; margin: 5px 0;">SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌Note You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

◆Important

If you are configuring notifications for an interface table, you must set the application context by specifying the requisite binding properties. For more information about setting the application context see [Setting Application Context](#).

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle E-Business adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-OracleEBS receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the Oracle database for Insert and Update operations. Configure this port to drop notification messages to the folder C:\TestLocation\UpsertNotification.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the Oracle database for all other operations. Configure this port to drop notification messages to the folder C:\TestLocation\OtherNotification.

Starting the Application

You must start the BizTalk application for receiving notification messages from the Oracle database and for performing the subsequent Select and Update operations. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleEBS one-way receive port, which receives the notification messages from the Oracle database is running.
- The two FILE send ports, which receive messages from Oracle database, are running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you start the BizTalk orchestration, the following set of actions take place:

- Because the **NotifyOnListenerStart** binding property is set to **True**, you receive the following message:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Info>ListenerStarted</Info>
  <Source>OracleEBSBinding</Source>
  <Type>Startup</Type>
</Notification>
```

Note that the value in the `<Info>` tag is "ListenerStarted". Hence, this message is received in C:\TestLocation\OtherNotification folder.

- Insert a record in the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
```

```

    <Info>1</Info>
    <QueryId>0</QueryId>
  </NotificationDetails>
</Details>
<Info>Insert</Info>
<ResourceNames>
  <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
</ResourceNames>
<Source>Data</Source>
<Type>Change</Type>
</Notification>

```

Note that the value in the `<Info>` tag is "Insert". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Update a record in the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>32</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
<Info>Update</Info>
<ResourceNames>
  <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
</ResourceNames>
<Source>Data</Source>
<Type>Change</Type>
</Notification>

```

Note that the value in the `<Info>` tag is "Update". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Delete a record from the ACCOUNTACTIVITY table. You will receive a notification message resembling the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Details>
    <NotificationDetails>
      <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
      <Info>16</Info>
      <QueryId>0</QueryId>
    </NotificationDetails>
  </Details>
<Info>Delete</Info>
<ResourceNames>
  <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
</ResourceNames>
<Source>Data</Source>
<Type>Change</Type>

```

```
</Notification>
```

Note that the value in the `<Info>` tag is "Delete". Hence, this message is received in C:\TestLocation\OtherNotification folder.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

Performing Complex Operations After Receiving Notification Messages

For simplicity and better understanding, the orchestration in this topic copies messages to different folders based on the notification type. However, in real-world scenarios you might want to perform more complex operations. You can perform similar procedures as provided in this topic and build on them to perform the operations you wish. For example, you can change the orchestration to insert records in another table if you get a notification message for an Insert operation on the ACCOUNTACTIVITY table. In such a case, you can make appropriate changes within the Decide shape.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Receiving Database Change Notifications Incrementally Using BizTalk Server

◆ Important

For the sake of brevity, this topic only describes how to receive notifications incrementally. In business scenarios, the orchestration must ideally include the logic to extract the kind of notification message received and then perform any subsequent operations. In other words, the orchestration described in this topic must be built on top of the orchestration described in [Processing Notification Messages to Perform Specific Tasks](#).

This topic demonstrates how to configure the Oracle E-Business adapter to receive incremental query notification messages from Oracle. To demonstrate incremental notifications, we consider a table, ACCOUNTACTIVITY, with a "Processed" column. When a new record is inserted to this table, the value of the "Processed" column is set to 'n'. You can configure the adapter to receive incremental notifications by doing the following:

- Register for notifications using a SELECT statement that retrieves all records that have "Processed" column as 'n'. You can do so by specifying the SELECT statement for the **NotificationStatement** binding property.
- For rows which have been notified for, update the "Processed" column to 'y'.

This topic demonstrates how to create a BizTalk orchestration and configure a BizTalk application to achieve this.

Configuring Notifications with the Oracle E-Business Adapter Binding Properties

The table below summarizes the Oracle E-Business adapter binding properties that you use to configure receiving notifications from Oracle E-Business Suite. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

📌 Note

You may choose to specify these binding properties when generating the schema for the **Notification** operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-OracleEBS receive port with the binding properties already set. For more information about creating a receive port using the binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the SELECT statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified SELECT statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to receive notifications from Oracle E-Business Suite, read further.

How This Topic Demonstrates Receiving Notification Messages

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving incremental database change notification

messages from Oracle E-Business Suite, we will configure the adapter to receive notifications for changes to the ACCOUNTACTIVITY table. Let us assume that the ACCOUNTACTIVITY table has columns "TID", "Account", and "Processed". Whenever a new record is added, the value of the "Processed" column is set to 'n'. So, to get incremental notifications you will have to do the following tasks as part of the BizTalk orchestration:

- Get notification for all records where "Processed" is 'n'. You can do this by specifying a SELECT statement as a notification statement.
- After the notification is received for a certain record, set "Processed" to 'y'. You can do this by executing a stored procedure, PROCESS_RECORDS, which updates the "Processed" column.

To demonstrate receiving incremental notifications, we do the following:

- Generate schema for the **Notification** (inbound operation), and **PROCESS_RECORDS** (outbound operation) on the ACCOUNTACTIVITY table.
- Create an orchestration that has the following:
 - A receive location to receive notification messages. You can configure for notification by specifying the SELECT statement as:

```
SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'
```

Note

You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.

- A send port to update the rows for which notification has already been sent. You will execute the PROCESS_RECORDS stored procedure on this port to set the value of "Processed" column to 'y' for the records for which notification is received.

Note that this operation must be executed after receiving the notification messages so that the processed rows are updated. To do away with the overhead of waiting to get the notification response and then manually dropping a request message to execute the PROCESS_RECORDS procedure, you will generate the request message for PROCESS_RECORDS procedure within the orchestration itself. You can do so by using the **Construct Message** shape within an orchestration.

How to Receive Notification Messages from Oracle E-Business Suite

Performing an operation on Oracle E-Business Suite using Oracle E-Business adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** (inbound operation) and **PROCESS_RECORDS** procedure (outbound operation) on the ACCOUNTACTIVITY table. Optionally, you can specify values for the **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from Oracle E-Business Suite.
3. Create messages in the BizTalk project for executing the PROCESS_RECORDS stored procedure and receiving response messages.
4. Create an orchestration that does the following:
 - Receives notification message from Oracle E-Business Suite.
 - Creates a message to execute the PROCESS_RECORDS procedure.

- Sends this message to Oracle E-Business Suite to select and update the records and receive a response.

5. Build and deploy the BizTalk project.
6. Configure the BizTalk application by creating physical send and receive ports.

Note

For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-OracleEBS receive port. Two-way receive ports are not supported for inbound operations.

7. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Notification** operation and **PROCESS_RECORDS** procedure. See [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **InboundOperationType**, **NotificationPort**, and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.
4. Select the contract type as **Client (Outbound operations)**.
5. Generate schema for the **PROCESS_RECORDS** procedure. This procedure is available under the **ACCOUNT_PKG** package.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create three messages—one to receive notifications from the Oracle E-Business Suite, one to execute the **PROCESS_RECORDS** procedure, and one to receive the response for the procedure.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>OracleNotifyIncremental.OracleEBSBinding.Notification</i> , where <i>OracleNotifyIncremental</i> is the name of your BizTalk project. <i>OracleEBSBinding</i> is the schema generated for the Notification operation.

6. Repeat step 3 to create two new messages. In the **Properties** pane for the new message, do the following:

Set Identifier to	Set Message Type to
Procedure	<i>OracleNotifyIncremental.OracleEBSBinding1.PROCESS_RECORDS</i> , where <i>OracleEBSBinding1</i> is the schema generated for the PROCESS_RECORDS procedure.
ProcedureResponse	<i>OracleNotifyIncremental.OracleEBSBinding1.PROCESS_RECORDSResponse</i>

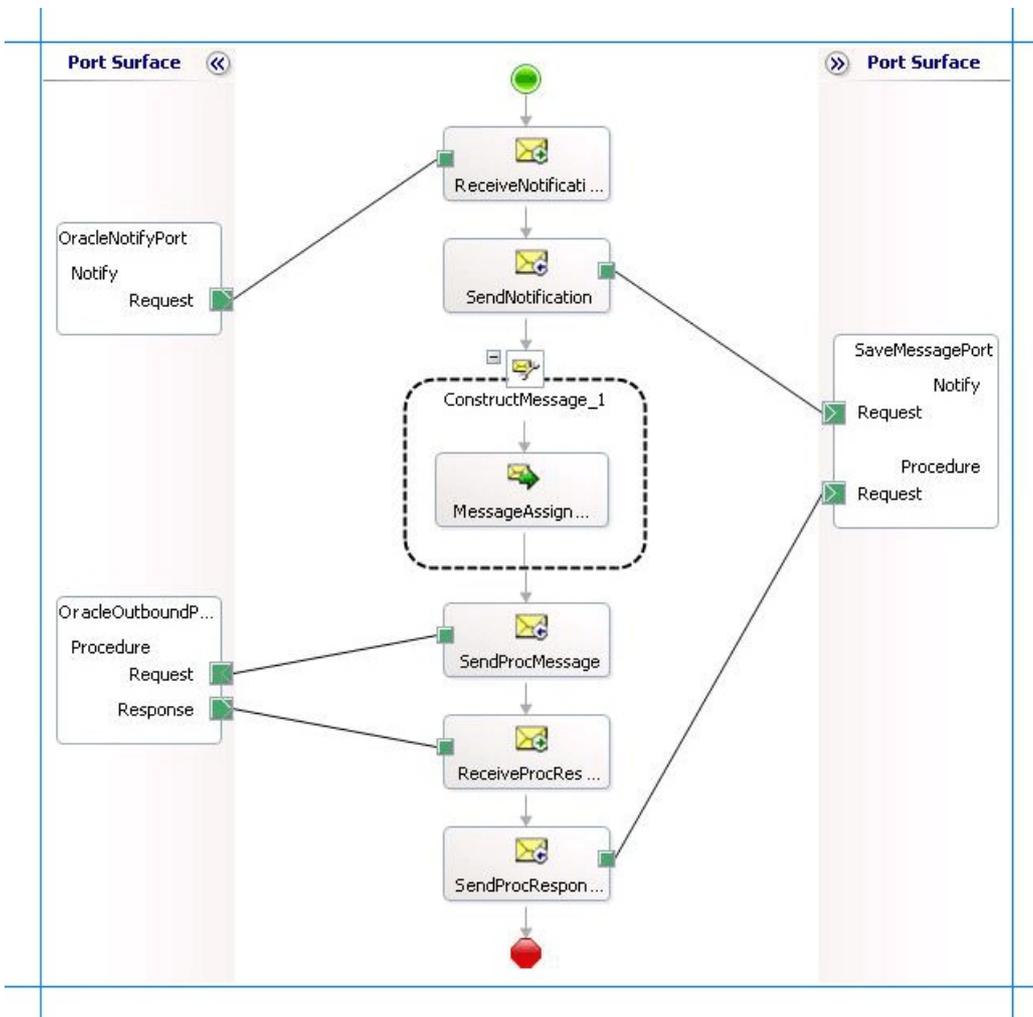
Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from Oracle E-Business Suite and then updating the rows for which notification was received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The notification message is received at a FILE location. Once the response is received, the orchestration constructs a message to invoke the PROCESS_RECORDS procedure, which updates the rows for which notification is received. The response for this message is also received at the same FILE location.

So, your orchestration must contain the following:

- A one-way WCF-Custom or WCF-OracleEBS receive port to receive notification messages.
- A two-way WCF-Custom or WCF-OracleEBS send port to send messages to execute the PROCESS_RECORDS procedure.
- A **Construct Message** shape to construct messages, to execute PROCESS_RECORDS procedure, within the orchestration.
- A FILE send port to save the notification message and the response for the PROCESS_RECORDS procedure.
- Receive and send shapes.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveNotification</i> Set Activate to <i>True</i>
SaveNotification	Send	<ul style="list-style-type: none"> Set Name to <i>SaveNotification</i>
SendProcMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendProcMessage</i>
ReceiveProcResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveProcResponse</i>
SaveProcResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SaveProcResponse</i>

Adding Construct Message Shape

You can use the **Construct Message** shape to generate a request message within the orchestration to execute the PROCESS_RECORDS procedure. To do so, you must add a **Construct Message** shape and within that a **Message Assignment** shape to your orchestration. For this example, the **Message Assignment** shape invokes code that generates a message that is sent to Oracle E-Business Suite to execute the procedure. The **Message Assignment** shape also sets the action for the message to be sent to Oracle E-Business Suite.

For the construct message shape, set the **Message Constructed** property to **Procedure**.

The code to generate the response could be part of the same Visual Studio solution as your BizTalk project. A sample code for

generating a response message looks like this.

```
namespace MessageCreator
{
    public class MessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;

        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\MessageIn";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOption.To
                pDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }
            //Create Message From XML
            Message = new XmlDocument();
            Message.PreserveWhitespace = true;
            Message.Load(ResponseDoc);
            return Message;
        }
    }
}
```

For the above code excerpt to be able to generate a request message, you must have an XML request message (for the PROCESS_RECORDS procedure) in the location specified for the `XmlFileLocation` variable.

Note

After you build the project, `MessageCreator.dll` will be created in the project directory. You must add this DLL to the global assembly cache (GAC). Also, you must add the `MessageCreator.dll` as a reference in the BizTalk project.

Add the following expression to invoke this code from the **Message Assignment** shape and to set the action for message. To add an expression, double-click the **Message Assignment** shape to open the Expression Editor.

```
Procedure = MessageCreator.MessageCreator.XMLMessageCreator();
Procedure(WCF.Action) = "PackageApis/SCOTT/ACCOUNT_PKG/PROCESS_RECORDS";
```

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
OracleNotifyPort	<ul style="list-style-type: none">Set Identifier to <i>OracleNotifyPort</i>Set Type to <i>OracleNotifyPortType</i>Set Communication Pattern to <i>One-Way</i>Set Communication Direction to <i>Receive</i>

SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i> • Create an operation <i>Notify</i>. This operation is used for notification messages. • Create an operation <i>Procedure</i>. This operation is used for select response messages.
OracleOutboundPort	<ul style="list-style-type: none"> • Set Identifier to <i>OracleOutboundPort</i> • Set Type to <i>OracleOutboundPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>OracleNotifyPort.Notify.Request</i>
SaveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>SaveMessagePort.Notify.Request</i>
SendProcMessage	<ul style="list-style-type: none"> • Set Message to <i>Procedure</i> • Set Operation to <i>OracleOutboundPort.Procedure.Request</i>
ReceiveProcResponse	<ul style="list-style-type: none"> • Set Message to <i>ProcedureResponse</i> • Set Operation to <i>OracleOutboundPort.Procedure.Response</i>
SaveProcResponse	<ul style="list-style-type: none"> • Set Message to <i>ProedureResponse</i> • Set Operation to <i>SaveMessagePort.Procedure.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-OracleEBS one-way receive port. This port listens for notifications coming from Oracle E-Business Suite. For information about how to create receive ports, see [Manually Configuring a Physical Port Binding to the Oracle E-Business Adapter](#). Make sure you specify the following binding properties for the receive port.

◆Important
 You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p>◆Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre>SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p>📌Note You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

◆Important

If you are configuring notifications for an interface table, you must set the application context by specifying the requisite binding properties. For more information about setting the application context see [Setting Application Context](#).

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the Oracle E-Business adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-OracleEBS receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a physical WCF-Custom or WCF-OracleEBS send port to send messages to Oracle E-Business Suite to execute the PROCESS_REOCRDS procedure. You must also specify the action in the send port.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the messages from Oracle E-Business Suite. These will be the notification messages received from Oracle E-Business Suite and messages for the PROCESS_RECORDS procedure you execute through the WCF-Custom or WCF-OracleEBS send port.

Starting the Application

You must start the BizTalk application for receiving notification messages from Oracle E-Business Suite and for executing the PROCESS_RECORDS procedure. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-OracleEBS one-way receive port, which receives the notification messages from Oracle E-Business Suite is running.
- The WCF-Custom or WCF-OracleEBS send port to execute the PROCESS_RECORDS procedure is running.
- The FILE send port, which receives messages from Oracle E-Business Suite, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

Assume that the ACCOUNTACTIVITY table already has some records. Also, make sure the XML message to execute PROCESS_RECORDS procedure is available at C:\TestLocation\Messageln. The XML file should resemble the following:

```
<PROCESS_RECORDS xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/SCOTT/ACCOUNT_PKG" />
```

Once the BizTalk orchestration is started, the following set of actions take place, in the same sequence:

- The adapter receives a notification message that resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">
  <Info>ListenerStarted</Info>
  <Source>OracleEBSBinding</Source>
  <Type>Startup</Type>
</Notification>
```

This message notifies that the receive port for receiving the notification messages is started. Note that the value for the <Info> element is "ListnerStarted".

- The adapter executes the PROCESS_RECORDS procedure. The next response from Oracle E-Business Suite is for the procedure.

```

<?xml version="1.0" encoding="utf-8" ?>
<PROCESS_RECORDSResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Package
Apis/SCOTT/ACCOUNT_PKG">
  <TABLE_DATA>
    <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdat
a="urn:schemas-microsoft-com:xml-msdata">
      <xs:element msdata:IsDataSet="true" name="NewDataSet">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                  <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
      <NewDataSet xmlns="">
        <NewTable>
          <TID>1</TID>
          <ACCOUNT>100001</ACCOUNT>
          <PROCESSED>n</PROCESSED>
        </NewTable>
        <NewTable>
          .....
          .....
        </NewTable>
        .....
        .....
      </NewDataSet>
    </diffgr:diffgram>
  </TABLE_DATA>
</PROCESS_RECORDSResponse>

```

This is the response for the SELECT statement execute as part of the PROCESS_RECORDS procedure.

- The PROCESS_RECORDS procedure also updates the rows to set PROCESSED to 'y'. Hence, the adapter receives another notification for the Update operation.

```

<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/">

```

```

<Details>
  <NotificationDetails>
    <ResourceName>SCOTT.ACCOUNTACTIVITY</ResourceName>
    <Info>32</Info>
    <QueryId>0</QueryId>
  </NotificationDetails>
</Details>
<Info>Update</Info>
<ResourceNames>
  <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">SCOTT.AC
COUNTACTIVITY</string>
</ResourceNames>
<Source>Data</Source>
<Type>Change</Type>
</Notification>

```

Note that the `Info` element contains "Update".

- After the second notification, the adapter again executes the `PROCESS_RECORDS` procedure. However, now because there are no records where `PROCESSED` column is set to 'n', the procedure returns an empty response resembling the following.

```

<?xml version="1.0" encoding="utf-8" ?>
<PROCESS_RECORDSResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Package
Apis/SCOTT/ACCOUNT_PKG">
  <TABLE_DATA>
    <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdat
a="urn:schemas-microsoft-com:xml-msdata">
      <xs:element msdata:IsDataSet="true" name="NewDataSet">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" name="TID" type="xs:decimal" />
                  <xs:element minOccurs="0" name="ACCOUNT" type="xs:decimal" />
                  <xs:element minOccurs="0" name="PROCESSED" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
    <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
      <NewDataSet xmlns="" />
    </diffgr:diffgram>
  </TABLE_DATA>
</PROCESS_RECORDSResponse>

```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

Receiving Database Change Notifications On Multiple Receive Locations

Consider a scenario where you have multiple receive locations created as part of different BizTalk applications configured to receive query notifications for the same table (e.g. ACCOUNTACTIVITY) in the same database. If a hundred records are inserted into the same table, all the receive locations will get the notification message. To effectively receive notifications across multiple receive locations, you can call operations from your BizTalk application in such a way that if a notification is received by one receive location, the other receive location does not get the same notification. So, you can effectively load-balance notifications received on multiple locations.

The tasks required to set up an orchestration to load-balance receiving notifications are same as that for [Receiving Database Change Notifications Incrementally Using BizTalk Server](#). This topic lists the only the difference between the two approaches.

Load-Balancing Query Notifications Across Multiple Receive Locations

Like in the topic [Receiving Database Change Notifications Incrementally Using BizTalk Server](#), you configured incremental notifications by executing a PROCESS_RECORDS procedure. To configure load-balancing, you could execute a stored procedure that deletes the records that have been notified for. For example, consider a stored procedure NOTIFY_LOAD_BALANCE with the following definition:

```
PROCEDURE NOTIFY_LOAD_BALANCE (TABLE_DATA OUT SYS_REFCURSOR) IS
  var int;
BEGIN
  SELECT TID INTO var FROM ACCOUNTACTIVITY WHERE ROWNUM = 1 FOR UPDATE;
  OPEN TABLE_DATA FOR SELECT * FROM ACCOUNTACTIVITY WHERE TID = var;
  DELETE FROM ACCOUNTACTIVITY WHERE TID = var;
END NOTIFY_LOAD_BALANCE;
```

When you execute this stored procedure as part of the BizTalk application, the record for which notification is already received gets deleted. So, the other receive location gets notification for the next record.

Here are the high-level steps you must perform to configure load-balancing for receiving notifications.

1. Create schema for **Notification** (inbound operation) and **NOTIFY_LOAD_BALANCE** procedure (outbound operation).
2. Add an orchestration and add three messages for receiving notification, executing the procedure, and getting response for the procedure.
3. Create an orchestration by adding Send and Receive shapes, Construct Message shape, and ports. You can use the same sample code for constructing a message to invoke the NOTIFY_LOAD_BALANCE stored procedure. Note that while performing the operation in BizTalk Server Administration console, you must have the request message for the NOTIFY_LOAD_BALANCE procedure in the location C:\TestLocation\MessageIn. You do so because the code snippet you invoke as part of the orchestration created in [Receiving Database Change Notifications Incrementally Using BizTalk Server](#) creates a request message based on the request XML present in C:\TestLocation\MessageIn.
4. Build and deploy the application. To demonstrate load-balancing, you must deploy this orchestration at least on two different computers that have BizTalk Server and Oracle E-Business adapter installed.
5. In the BizTalk Server Administration console on both the computers, specify the following binding properties for the WCF-Custom or WCF-OracleEBS receive location:

Binding Property	Value

InboundOperationType	Set this to Notification .
NotificationPort	<p>Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database. Set this to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see http://go.microsoft.com/fwlink/?LinkID=196959.</p> <p> Important If you set this to the default value of -1, you will have to completely disable Windows Firewall to receive notification messages.</p>
NotificationStatement	<p>Set this to:</p> <pre>SELECT TID,ACCOUNT,PROCESSED FROM SCOTT.ACCOUNTACTIVITY WHERE PROCESSED = 'n'</pre> <p> Note You must specify the table name along with the schema name. For example, SCOTT.ACCOUNTACTIVITY.</p>
NotifyOnListenerStart	Set this to True .

6. Start the BizTalk application.
7. To start receiving notifications, insert a hundred records into the ACCOUNTACTIVITY table. While doing so, make sure the request XML for invoking the NOTIFY_LOAD_BALANCE procedure is available in C:\TestLocation\MessageIn.
8. Monitor the location (on both the computers) where the BizTalk application will be dropping the notification messages. You will notice that of the hundred records inserted, one location gets notifications for some records while the other location gets notification for the remaining records. Together, both the locations will get notification for all the hundred records.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Receiving Database Change Notifications After a Receive Location Breakdown

Consider a scenario where you have a BizTalk application that receives database change notification messages when changes are made to the ACCOUNTACTIVITY table. If the receive location configured as part of the BizTalk application breaks down, and simultaneously records are added into the ACCOUNTACTIVITY table, you will not receive notifications for the recently added records. You will also not know when the receive location is available again. The Oracle E-Business adapter exposes a binding property, **NotifyOnListenerStart**, that you can configure to get a notification that the receive location has recovered. You can specify the following values for the **NotifyOnListenerStart** binding property:

- Set this property to **True**, to receive a notification informing that the receive location is available, as soon as the receive location recovers.
- Set this property to **False**, to not receive a notification informing that the receive location is available, after the receive location recovers.

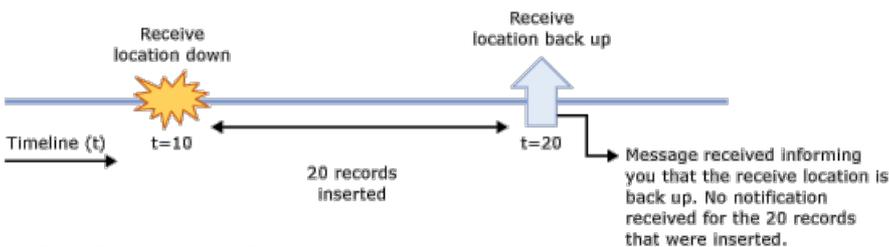
Default is **True**.

Configuring the Oracle E-Business Adapter Behavior

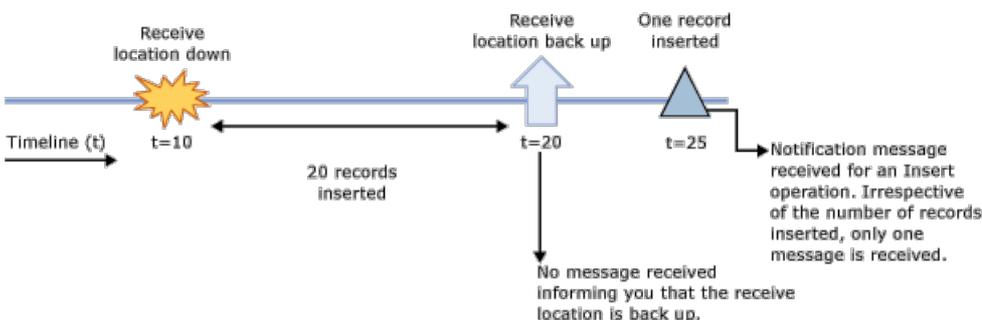
For either of the approaches, you do not need to perform any specific tasks while generating metadata or while configuring the BizTalk application. You only need to set the **NotifyOnListenerStart** binding property accordingly on the WCF-Custom or WCF-OracleEBS receive location. To create the BizTalk application, you must perform the same set of tasks as described in [Receiving Database Change Notifications Incrementally Using BizTalk Server](#). However, when configuring the BizTalk application using BizTalk Server, you can try changing the value of **NotifyOnListenerStart** binding property and see the difference in the two configurations.

The following figure demonstrates how the notifications are received based on the value of the **NotifyOnListenerStart** binding property.

NotifyOnListenerStart = True



NotifyOnListenerStart = False



Note that in the first scenario, when the **NotifyOnListenerStart** is set to **True** and records are inserted into the database table while the receive location was down, the adapter only sends you a notification message when the receive location comes up. The adapter does not perform any operation to process the records that were inserted while the receive location was down. The adapter client must implement the relevant logic in their application to process the records that were inserted while the receive location was down.

See Also

Other Resources

[Receiving Database Change Notifications Using BizTalk Server](#)

Developing Applications By Using the WCF Service Model

Windows Communication Foundation (WCF) provides a programming model called the WCF service model to connect to the Microsoft BizTalk Adapter for Oracle E-Business Suite. The service model was added to WCF to address, in part, some of the limitations of the WCF channel programming model.

The WCF service model uses familiar .NET paradigms to hide the complexities of exchanging SOAP message over a channel. The service model accomplishes this simplification by reading the entire SOAP message into memory before copying the information into .NET data structures. Loading long messages into memory may not be practical for some applications. In these cases, developers should use the WCF channel model. For more information about using the WCF channel model, see [Developing Applications by Using the WCF Channel Model](#).

At the lowest level, the WCF presents the WCF channel model in which clients invoke operations on a service by exchanging SOAP messages over a channel established between client and service endpoints. The WCF channel model exposes data types and methods that enable you to operate directly on the WCF channel architecture. The WCF channel model provides you with direct control over the contents of the SOAP messages you create and over the way both your application and the Oracle E-Business adapter consume them. However, creating well-formed SOAP messages to send over a channel and validating the reply messages returned can be a detailed and exacting task.

The WCF service model uses proxy classes to invoke operations on a target service or to receive operations from a client. The Oracle E-Business adapter exposes the Oracle E-Business Suite as a WCF service on which you can invoke operations.

- The proxy class used to invoke operations on a target service is called a WCF client class. This class models the operations exposed by a service as .NET methods with strongly-typed parameters. By using the WCF service model, you can invoke the operations exposed by the Oracle E-Business adapter as .NET methods on the WCF client. For more information about WCF clients, see "WCF Client Overview" at <http://go.microsoft.com/fwlink/?LinkId=91458>

You can use either of the following tools to generate a WCF client class and associated helper code from the service metadata that the Oracle E-Business adapter exposes:

- **The ServiceModel Metadata Utility Tool (svcutil.exe)**, which ships with WCF.
- **The Add Adapter Service Reference Visual Studio Plug-in**, which ships with the Oracle E-Business adapter and is integrated with the Visual Studio design experience. This tool presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations that the adapter exposes. For more information about how to generate a WCF client, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

In This Section

The following topics give information about how to develop applications that use the WCF service model:

- [Overview of Using the WCF Service Model with the Adapter](#)
- [Metadata and the WCF Service Model](#)
- [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#)
- [Specifying a Client Binding for the Oracle E-Business Suite](#)
- [Performing Insert, Update, Delete, or Select Operations on Interface Tables and Views Using the WCF Service Model](#)
- [Performing Operations on Tables with Large Data Types Using the WCF Service Model](#)
- [Invoking Concurrent Programs in Oracle E-Business Suite Using the WCF Service Model](#)
- [Invoking Request Sets in Oracle E-Business Suite Using the WCF Service Model](#)

- [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using WCF Service Model](#)
- [Polling Oracle E-Business Suite Using the WCF Service Model](#)
- [Receiving Database Change Notifications Using the WCF Service Model](#)

See Also

Other Resources

[Development](#)

Overview of Using the WCF Service Model with the Adapter

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes an Oracle E-Business Suite system as a WCF service. To perform operations on Oracle E-Business Suite artifacts, for example to invoke a stored procedure, you invoke an operation on the adapter, which, in turn, performs the operation on the Oracle E-Business Suite. Your code acts as a client to the WCF service presented by the adapter.

In the Windows Communication Foundation (WCF) service model, the service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The Oracle E-Business adapter and WCF provide tools that enable you to generate this interface for targeted operations from the metadata that the adapter exposes. These tools also create a WCF client class that can be used to invoke the operations exposed in the service interface. A client application can call the methods of the WCF client class to invoke operations on the adapter.

The following section explains how to use the WCF service model to invoke operations with a WCF client.

Invoking Operations on the Oracle E-Business Suite with a WCF Client

To use the WCF service model to invoke operations on the Oracle E-Business adapter, you must first generate a WCF client class for the target operations. You can then create an instance of this class, a WCF client, and call its methods to perform these operations on the Oracle E-Business Suite.

To invoke operations on the Oracle E-Business adapter

1. Generate a WCF client class and helper code. Use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class targeted at Oracle E-Business Suite artifacts with which you want to work. For more information about how to generate a WCF client, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).
2. Create a WCF client instance and configure the WCF client. Configuring the WCF client involves specifying the binding and endpoint address (connection URI) that the client will use. You can do this either imperatively in code or declaratively in configuration. The following code creates a WCF client that targets the **Customer Interface** concurrent program in the **Receivables** application in the Oracle E-Business Suite. It also sets the credentials for the Oracle E-Business Suite. The WCF client is initialized from configuration.

```
ConcurrentPrograms_ARClient client = new ConcurrentPrograms_ARClient("OracleEBSBinding
_ConcurrentPrograms_AR"); //picking the binding and address from app.config

client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

Note

You can either specify the client binding and endpoint address in the code or declare it in the app.config configuration file. The preceding code snippet uses the latter. For more information on how to use either approaches, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

3. Open the WCF client.

```
client.Open();
```

4. Invoke methods on the WCF client created in step 2 to perform operations on the Oracle E-Business Suite. The following code invokes the **Customer Interface** concurrent program in the **Receivables** application in the Oracle E-Business Suite.

```
string Result = client.RACUST(null, null, null, description, null, recipro_cust, org_id);
```

RACUST is the actual name of the Customer Interface concurrent program. **Customer Interface** is the friendly name of the concurrent program.

5. Close the WCF client.

```
client.Close();
```

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Metadata and the WCF Service Model

In the WCF service model, you use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutile.exe) to do the following:

- Generate a service contract—the WCF service contract—through which your code can receive operations from the adapter. This .NET interface represents the service contract for target operations.
- Generate proxy classes—the WCF client class—through which your code can invoke operations on the adapter.
- Annotated classes that represent the supporting message contracts, operation contracts, and data contracts for the service contract.

For help in understanding the structure of this generated code, see "Understanding Generated Client Code" at <http://go.microsoft.com/fwlink/?LinkId=98365>. This topic specifically describes code that svcutil.exe generates, but its content is also applicable to the code that the Add Adapter Service Reference Plug-in generates.

For information about how to generate a WCF client class or WCF service contract for target operations and about the differences between svcutil.exe and the Add Adapter Service Reference Plug-in, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts

You can use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class or a WCF service contract (interface) targeted at selected operations on Oracle E-Business Suite artifacts. You can also use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate the WCF client class or WCF service contract; however, the Add Adapter Service Reference Plug-in exposes the functionality of the ServiceModel Metadata Utility Tool through a standard Microsoft Windows interface. It also provides browse and search capabilities that are not available with the svcutil.exe tool, and it generates a configuration file based on the binding properties that you select when you connect to the Oracle E-Business Suite.

Generating a Client Class by Using the Add Adapter Service Reference Plug-in

Perform the following steps to generate a WCF client class by using the Add Adapter Service Reference Plug-in.

To generate a WCF client class

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) to connect to the Oracle E-Business Suite and browse and search for operations. To create a WCF client class for the operations that you select, be sure that **Client (Outbound operations)** is selected from the **Select contract type** drop-down list (this is the default).
3. After you select all of the operations that you want to target, click **OK** to generate the WCF client class.

The Add Adapter Service Reference Plug-in adds two files to your project:

- **OracleEBSBindingClient.cs**. This file contains the generated WCF client class and helper code for the operations that you selected.
- **app.config**. This file contains a binding configuration and client endpoint configurations. These configurations are based on the selections you made when you configured the binding and connection for the Add Adapter Service Reference Plug-in.

Important

While using the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Generating a WCF Service Contract by Using the Add Adapter Service Reference Plug-in

The adapter exposes inbound operations to enable Oracle E-Business Suite to send messages to an adapter client. For such operations you must generate a WCF service contract. This section provides information on how to generate a service contract for inbound operations exposed by the adapter.

Perform the following steps to generate a WCF service contract by using the Add Adapter Service Reference Plug-in.

To generate a WCF service contract for inbound operations

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for Oracle E-Business Suite Operations in Visual Studio](#) to connect to the Oracle E-Business Suite. There are several binding properties and a URI property that you may want to set when you connect to the Oracle E-Business Suite.
3. After you have connected to the Oracle E-Business Suite, select **Service (Inbound operations)** from the **Select contract type** drop-down list.
4. In the **Select a category** box, browse to the inbound operation for which you want to generate the service contract. For example, for **Notification** operation, click the root node (*/*), select **Notification** from the **Available categories and operations** box, and then click **Add**. For instructions on how to browse for inbound operations, see

5. To generate the WCF service contract for the operation, click **OK**.

The Add Adapter Service Reference Plug-in adds three files to your project:

- **OracleEBSBindingInterface.cs**. This file contains the generated WCF service contract (interface) and helper code for the inbound operation.
- **OracleEBSBindingService.cs**. This file contains a class that implements the interface defined in OracleDBBindingInterface.cs. You can implement the business logic that processes the records returned by the inbound operation.
- **app.config**. This file contains a binding configuration, endpoint behaviors, and service endpoint configuration that are based on the selections you made when you configured the binding and connection for the Add Adapter Service Reference Plug-in.

◆ Important

While using the Add Adapter Service Reference Visual Studio Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Using svcutil.exe to Generate a WCF Client Class or a WCF Service Contract

You can use svcutil.exe to generate a WCF client class or a WCF service interface for your application. You must configure svcutil.exe to use it with the Oracle E-Business adapter.

Svcutil.exe generates the WCF client class or WCF service contract in an output file. The default file name is output.cs. You must manually include this file in your Visual Studio project. For more information about svcutil.exe, see <http://go.microsoft.com/fwlink/?LinkId=139432>.

Specifying a Client Binding for the Oracle E-Business Suite

After you have generated the WCF client class, you can create a WCF client (instance) and invoke its methods to consume the Oracle E-Business adapter.

To create the WCF client, you must specify an endpoint address and a binding. The endpoint address must contain a valid Oracle E-Business Suite connection URI, and the binding must be an instance of an Oracle E-Business Suite binding (**OracleEBSBinding**). For more information about the Oracle E-Business Suite connection URI, see [Establishing a Connection to the Oracle E-Business Suite](#). We recommend that you do not specify the user credentials as part of the connection URI. You may instead use the **ClientCredentials** property of the WCF client, as explained in this topic.

You can specify the Oracle E-Business Suite binding and the endpoint address in your code or in a configuration file. When you use the Add Adapter Service Reference Visual Studio Plug-in to generate the WCF client class, a configuration file (app.config) is also created for your project. This file contains configuration settings that reflect the binding properties and connection information (except credentials) that you specified when you connected to the Oracle E-Business Suite with the Add Adapter Service Reference Plug-in.

Specifying the Binding and Endpoint Address in Code

The following code shows how to create a WCF client by specifying the binding and endpoint address in code using the **ClientCredentials** property of the WCF client.

```
//Create an Oracle EBS binding and set the binding properties
OracleEBSBinding binding = new OracleEBSBinding();
binding.OracleUserName = "myOracleEBSUser";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "Responsibility";
binding.OracleEBSOrganizationId = "204";

//Create the client endpoint
EndpointAddress address = new EndpointAddress("oracleebs://<oracleebs_instance_name>");

//Create the client and specify the credentials
ConcurrentPrograms_ARClient client = new ConcurrentPrograms_ARClient(binding,address);
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";

//Open the client
client.Open();
```

Specifying the Binding and Endpoint Address in a Configuration File

The following code shows how to create a WCF client by specifying the binding and endpoint address in an app.config file.

```
//Create the client by specifying the endpoint name in the app.config. In this case, the binding properties
//must also be specified in the app.config.
ConcurrentPrograms_ARClient client = new ConcurrentPrograms_ARClient("OracleEBSBinding_ConcurrentPrograms_AR");

//Specify the credentials
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";

client.Open();
```

The following XML shows the configuration file created for the **Customer Interface** concurrent program by the Add Adapter Service Reference Plug-in. This file contains the client endpoint configuration referenced in the preceding example.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
```

```

    <oracleEBSBinding>
      <binding openTimeout="00:05:00" name="OracleEBSBinding" closeTimeout="00:01
:00"
      Database"
        receiveTimeout="00:10:00" sendTimeout="00:01:00" clientCredentialType="
ge="false"
        inboundOperationType="Polling" metadataPooling="true" statementCachePur
statementCacheSize="10" pollWhileDataFound="false" pollingInterval="30"
useOracleConnectionPool="true" minPoolSize="1" maxPoolSize="100"
incrPoolSize="5" decrPoolSize="1" connectionLifetime="0" acceptCredenti
alsInUri="false"
        useAmbientTransaction="true" notifyOnListenerStart="true"
        notificationPort="-1" dataFetchSize="65536" longDatatypeColumnSize="325
12"
        skipNilNodes="true" maxOutputAssociativeArrayElements="32"
        enableSafeTyping="false" insertBatchSize="20" useSchemaInNameSpace="tru
e"
        enableBizTalkCompatibilityMode="true">
        <mlsSettings language="" dateFormat="" dateLanguage="" numericCharacter
s=""
        sort="" territory="" comparison="" currency="" dualCurrency=""
        iSOCurrency="" calendar="" lengthSemantics="" nCharConversionExcept
ion="true"
        timeStampFormat="" timeStampTZFormat="" timeZone="" />
      </binding>
    </oracleEBSBinding>
  </bindings>
<client>
  <endpoint address="oracleEBS://EBS-70-12/" binding="oracleEBSBinding"
    bindingConfiguration="OracleEBSBinding" contract="ConcurrentPrograms_AR"
    name="OracleEBSBinding_ConcurrentPrograms_AR" />
</client>
</system.ServiceModel>
</configuration>

```

If a project has more than one WCF client, there will be multiple client endpoint entries defined in the configuration file. Each WCF client entry will have a unique name based on its binding configuration and target Oracle E-Business Suite artifact. If you connect multiple times to create the WCF clients in your project, multiple binding configuration entries will be created, one for each connection. These binding configuration entries will be named in the following manner: OracleEBSBinding, OracleEBSBinding1, and so on. Each client endpoint entry created during a specific connection will reference the binding entry created during that connection.

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Performing Insert, Update, Delete, or Select Operations on Interface Tables and Views Using the WCF Service Model

The Oracle E-Business adapter discovers a set of basic Insert, Select, Update, and Delete operations on interface tables. By using these operations, you can perform simple Insert, Select, Update, and Delete statements qualified by a WHERE clause on a target interface table. This topic provides instructions on how to perform these operations using the WCF service model.

Note

The Oracle E-Business adapter supports only Select operations on interface views.

For more information about how the adapter supports these operations, see [Operations on Interface Tables and Interface Views](#).

About the Examples Used in this Topic

The example in this topic performs operations on the MS_SAMPLE_EMPLOYEE interface table. The table is created by running the script provided with the samples. For more information about samples, see [Samples](#). A sample, **Interface_Table_Ops**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Client Class

The name of the WCF client generated for the basic operations that the Oracle E-Business adapter discovers is based on the name of the table or view, as listed in the following table.

Artifact	WCF Client Name
Interface tables	InterfaceTables_[APP_NAME]_[SCHEMA]_[TABLE_NAME]Client
Interface views	InterfaceViews_[APP_NAME]_[SCHEMA]_[VIEW_NAME]Client

[APP_NAME] = Actual name of the Oracle E-Business Suite application; for example, FND.

[SCHEMA] = Collection of artifacts; for example, APPS.

[TABLE_NAME] = The name of the table; for example, MS_SAMPLE_EMPLOYEE.

[VIEW_NAME] = The name of the view; for example, MS_SAMPLE_EMPLOYEE_View.

Method Signature for Invoking Operations on Tables

The following table shows the method signatures for the basic operations on an table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Insert	string Insert(InsertRecord[] RECORDSET);
Select	SelectRecord[] Select(string COLUMN_NAMES, string FILTER);
Update	string Update(UpdateRecord RECORDSET, string FILTER);
Delete	string Delete(string FILTER);

As an example, the following code shows the method signatures for a WCF client class generated for the Delete, Insert, Select and Update operations on the MS_SAMPLE_EMPLOYEE interface table under the default APPS schema.

```
public partial class InterfaceTables_FND_APPMS_MS_SAMPLE_EMPLOYEEClient : System.ServiceModel
1.ClientBase<InterfaceTables_FND_APPMS_MS_SAMPLE_EMPLOYEE>, InterfaceTables_FND_APPMS_SAMP
LE_EMPLOYEE {
    public SelectRecord[] Select(string COLUMN_NAMES, string FILTER);
```

```

public string Insert(InsertRecord[] RECORDSET);

public string Update(UpdateRecord RECORDSET, string FILTER);

public string Delete(string FILTER);
}

```

In this snippet, **InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEEClient** is the name of the WCF class in the OracleEBSBindingClient.cs generated by the Add Adapter Service Reference Plug-in.

Parameters for Table Operations

This section provides the parameters required by each table operation

Select Operation

COLUMN_NAMES	FILTER
A comma-delimited list of column names in the target; for example, "EMP_NO, DESIGNATION". The column list specifies the columns of the target that should be returned in the result set. Columns not specified in the column list will be set to their .NET default values in the returned record set. For nullable columns, this value is null .	The contents of a WHERE clause that specifies the target rows of the query; for example, "Designation = 'Manager'". You can set this parameter to null to return all rows of the target.

The return value for a Select operation is a strongly-typed result set that contains the specified columns and rows.

Insert Operation

Insert operation type	RECORDSET
Multiple record	A collection of INSERTRECORDS that should be inserted into the table.

The return value for an Insert operation is the number of rows inserted.

Update Operation

RECORDSET	FILTER
A collection of records that should be updated in the table.	The contents of a WHERE clause that specifies the target rows of the query; for example, "Designation = 'Manager'". You can set this parameter to null to return all rows of the target.

The return value for an Update operation is the number of rows updated.

Delete Operation

The Delete operation takes as input a WHERE clause that specifies the rows to be deleted. The return value for a Delete operation is the number of rows deleted.

Creating a WCF Client to Invoke Operations on Interface Tables and Interface Views

The generic set of actions required to perform an operation on Oracle E-Business Suite using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke basic Insert, Select, Update, Delete operations on an interface table.

To create a WCF client to perform operations on tables

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the Insert, Select, Update, and Delete operation on the MS_SAMPLE_EMPLOYEE interface table. For more information about generating a WCF client class, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS` and `Microsoft.ServiceModel.Channels`.

4. Open the Program.cs file and add the following namespaces:

- `Microsoft.Adapters.OracleEBS`
- `System.ServiceModel`

5. Open the Program.cs file and create a client as described in the snippet below.

```
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs_instance_name");
InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEEClient client = new InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEEClient(binding, address);
```

In this snippet, `InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEEClient` is the WCF client defined in `OracleEBSBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in.

Note

In this snippet, you explicitly specify the binding and endpoint address in your application code. You can use these values from the application configuration file, `app.config`, also generated by the Add Adapter Service Reference Plug-in. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

6. Set the credentials for the client.

```
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

7. Because you are performing an operation on an interface table, you must set the application context. In this example, to set the application context, you specify the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

```
binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";
```

8. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

9. Invoke the Insert operation on the `MS_SAMPLE_EMPLOYEE` table.

```
Console.WriteLine("The application will insert a record in the MS_SAMPLE_EMPLOYEE interface table");

// The date values cannot contain time zone information. Hence, you must use
// DateTimeKind.Unspecified to not include the time zone information.
```

```

DateTime date = new DateTime(DateTime.Now.Ticks, DateTimeKind.Unspecified);

string result;

InsertRecord[] recordSet = new InsertRecord[1];

EMP_NO__COMPLEX_TYPE emp_no = new EMP_NO__COMPLEX_TYPE();
emp_no.Value = "10007";

NAME__COMPLEX_TYPE name = new NAME__COMPLEX_TYPE();
name.Value = "John Smith";

DESIGNATION__COMPLEX_TYPE desig = new DESIGNATION__COMPLEX_TYPE();
desig.Value = "Manager";

SALARY__COMPLEX_TYPE salary = new SALARY__COMPLEX_TYPE();
salary.Value = "500000";

JOIN_DATE__COMPLEX_TYPE doj = new JOIN_DATE__COMPLEX_TYPE();
doj.Value = date;

recordSet[0] = new InsertRecord();
recordSet[0].EMP_NO = emp_no;
recordSet[0].NAME = name;
recordSet[0].DESIGNATION = desig;
recordSet[0].SALARY = salary;
recordSet[0].JOIN_DATE = doj;

try
{
    result = client.Insert(recordSet);
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("Number of records inserted= " + result);
Console.WriteLine("Press any key to continue...");
Console.ReadLine();

```

You can replace the preceding code snippet to perform Select, Update, or Delete operations as well. You can also append the code snippets to perform all operation in a single application. For code snippets on how to perform these operations, see [Select Operation](#), [Update Operation](#), and [Delete Operation](#) respectively.

10. Close the client as described in the snippet below:

```

client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();

```

11. Build the project and then run it. The application inserts a record in the MS_SAMPLE_EMPLOYEE table.

Select Operation

The following code shows a Select operation that targets the MS_SAMPLE_EMPLOYEE interface table. The Select operation selects the last record inserted into the table. The returned record is written to the console.

```

Console.WriteLine("The application will now select the last inserted record");
SelectRecord[] selectRecords;
try
{
    selectRecords = client.Select("?", "WHERE EMP_NO LIKE 10007");
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("The details of the newly added employee are:");
Console.WriteLine("*****");
for (int i = 0; i < selectRecords.Length; i++)
{
    Console.WriteLine("Employee ID      : " + selectRecords[i].EMP_NO);
    Console.WriteLine("Employee Name    : " + selectRecords[i].NAME);
    Console.WriteLine("Employee Designation : " + selectRecords[i].DESIGNATION);
    Console.WriteLine("Employee Salary    : " + selectRecords[i].SALARY);
    Console.WriteLine();
}
Console.WriteLine("*****");
Console.WriteLine("Press any key to continue ...");
Console.ReadLine();

```

Update Operation

The following code shows an Update operation that targets the MS_SAMPLE_EMPLOYEE interface table.

```

Console.WriteLine("The application will now update the employee name in the newly inserted
record");
string recordsUpdated;
UpdateRecord updateRecordSet = new UpdateRecord();
updateRecordSet.NAME = "Tom Smith";
updateRecordSet.DESIGNATION = "Accountant";

try
{
    recordsUpdated = client.Update(updateRecordSet, "WHERE EMP_NO LIKE 10007");
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("No of records updated: " + recordsUpdated);
Console.WriteLine("Press any key to continue...");
Console.ReadLine();

```

Delete Operation

The following code shows a Delete operation that targets the MS_SAMPLE_EMPLOYEE interface table.

```

Console.WriteLine("The sample will now delete the record that it first inserted");
string deletedRecords;
try
{
    deletedRecords = client.Delete("WHERE EMP_NO LIKE 10007");
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
Console.WriteLine("No of records deleted: " + deletedRecords);

```

```
Console.WriteLine("Press any key to exit...");  
Console.ReadLine();
```

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Performing Operations on Tables with Large Data Types Using the WCF Service Model

The Oracle E-Business adapter enables adapter clients to perform operations on interface tables and views with large data types such as BLOB, CLOB, NCLOB, and BFILE.

- For columns of type BLOB, CLOB, and NCLOB, the adapter enables clients to read as well as update data. The adapter exposes `Read_<LOBColumnName>` and `Update_<LOBColumnName>` operations to read and update data respectively, where `<LOBColumnName>` is the name of column with large data type. If there is more than one column with large data type in a single interface table, the adapter exposes as many read and update operations for that interface table.
- For columns of type BFILE, adapter clients can only read data. The adapter exposes `Read_<LOBColumnName>` operation to read data from columns of BFILE type. If there is more than one column with large data type in a single interface table, the adapter exposes as many read operations for the interface table.

For more information about these operations, see [Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#).

About the Examples Used in this Topic

The example in this topic updates a BLOB column (PHOTO) in the CUSTOMER database table and then retrieves the data from the same column. The table is created by running the script provided with the samples. For more information about samples, see [Samples](#). A sample, **LargeDataTypes_ServiceModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

Note
This topic lists detailed tasks for updating and reading columns of large data types in a base database table. You must perform the same set of tasks for updating and reading columns of large data types in an interface table.

The WCF Client Class

The name of the WCF client generated for the operations on tables with large data types by the Oracle E-Business adapter is based on the name of the table, as listed in the following table.

Artifact	WCF Client Name
Interface tables	InterfaceTables_[APP_NAME]_[SCHEMA]_[TABLE_NAME]Client

[APP_NAME] = Actual name of the Oracle E-Business Suite application; for example, FND.

[SCHEMA] = Collection of artifacts; for example, APPS.

[TABLE_NAME] = The name of the table; for example, MS_SAMPLE_EMPLOYEE.

[VIEW_NAME] = The name of the view; for example, MS_SAMPLE_EMPLOYEE_View.

Method Signature for Invoking Operations on Tables

The following table shows the method signatures for the basic operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
<code>Update_<column_name></code>	<code>public void Update_<column_name>(string FILTER, byte[] DATA);</code>
<code>Read_<column_name></code>	<code>public System.IO.Stream Read_<column_name>(string FILTER);</code>

As an example, the following code shows the method signatures for a WCF client class generated for the `Update_PHOTO` and `Read_PHOTO` operations on the CUSTOMER database table under the APPS schema.

```
public partial class Tables_APPSCUSTOMERClient : System.ServiceModel.ClientBase<Tables_APP
```

```

S_CUSTOMER>, Tables_APPS_CUSTOMER {
    public void Update_PHOTO(string FILTER, byte[] DATA);
    public System.IO.Stream Read_PHOTO(string FILTER);
}

```

In this snippet, **Tables_APPS_CUSTOMERClient** is the name of the WCF class in the OracleEBSBindingClient.cs generated by the Add Adapter Service Reference Plug-in. Update_PHOTO and Read_PHOTO are methods that can be invoked to update and read columns of large data types in a table.

Parameters for Table Operations

This section provides the parameters required by the Update_<column_name> and Read_<column_name> operation.

Operation name	Parameters
Update_<column_name>	<p>Requires the following parameters:</p> <ul style="list-style-type: none"> • <code>string FILTER</code>. This parameter must contain the where clause that denotes the record for which data has to be updated. For example, "WHERE Name='Mindy Martin'". • <code>byte[] DATA</code>. Contains a byte array of data to be update in a column of large data type.
Read_<column_name>	<p>Requires the following parameters:</p> <ul style="list-style-type: none"> • <code>string FILTER</code>. This parameter must contain the where clause that denotes the record from which the data has to be read. For example, "WHERE Name='Mindy Martin'".

Creating a WCF Client to Invoke Operations on Tables with Columns of Large Data Types

The generic set of actions required to perform an operation on Oracle E-Business Suite using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke Update_PHOTO and Read_PHOTO operations on a CUSTOMER database table.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the Update_PHOTO and Read_PHOTO operations on the CUSTOMER database table. For more information about generating a WCF client class, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS` and `Microsoft.ServiceModel.Channels`, `System.Transactions`.
4. Open the Program.cs file and add the following namespaces:
 - `Microsoft.Adapters.OracleEBS`
 - `System.ServiceModel`
 - `System.Transactions`
 - `System.IO`

- Open the Program.cs file and create a client as described in the snippet below.

```
Tables_APPS_CUSTOMERClient client = new Tables_APPS_CUSTOMERClient("OracleEBSBinding_ables_APPS_CUSTOMER");

client.ClientCredentials.UserName.UserName = "<Enter user name here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `Tables_APPS_CUSTOMERClient` is the WCF client defined in `OracleEBSBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in.

Note

In this snippet, you use the binding and endpoint address from the configuration file `app.config`. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

- Set the credentials for the client.

```
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

Important

In this example, you are performing operations on a database table. However, if you are performing operations on an interface table, you must set the application context by specifying appropriate values for the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. You must specify these binding properties before opening the client. For more information about application context, see [Setting Application Context](#).

- Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

- Invoke the `Update_PHOTO` operation on the `CUSTOMER` table.

The `Update_PHOTO` operation requires a byte array for the data to be updated. In this code snippet, you use the `FileStream` class to create a byte array for a photo, `SamplePhoto.jpg`. For this application to work, the file must be copied to the project's `bin` directory.

Important

The `Update_PHOTO` operation must be performed in a transaction, so the **UseAmbientTransaction** binding property must be set to **true** and the `Update_PHOTO` operation must be performed within a transaction scope. You can set the **UseAmbientTransaction** binding property either in the `app.config` or by explicitly setting it in your application as `binding.UseAmbientTransaction = true`. Note that if you are specifying the binding property explicitly in the code, you must do so before opening the client.

```

byte[] photo;

using (FileStream fs = new FileStream("SamplePhoto.jpg", FileMode.Open))
{
    try
    {
        Console.WriteLine("Reading the photo");
        int count = 0;
        photo = new byte[fs.Length];
        while ((count += fs.Read(photo, count, (int)((fs.Length - count) > 4096) ? 40
96 : fs.Length - count))) < fs.Length) ;
    }
    catch(Exception ex)
    {
        Console.WriteLine("Exception: " + ex.Message);
        throw;
    }
}

Console.WriteLine("Updating data for the 'PHOTO' column");
// Invoking the Update_PHOTO operation inside a transaction scope
using (TransactionScope tx = new TransactionScope())
{
    string filter = "WHERE Name='Mindy Martin'";
    client.Update_PHOTO(filter, photo);
    tx.Complete();
}

```

9. Invoke the Read_PHOTO operation on the CUSTOMER table.

The Read_PHOTO gives the output in the form of System.IO.Stream. The adapter client must implement the FileStream class to read the data from Read_PHOTO operation. After the Read_PHOTO operation is complete, a file PhotoCopy.jpg is copied under the project's bin directory.

```

using (FileStream fs = new FileStream("PhotoCopy.jpg", FileMode.Create))
{
    Console.WriteLine("Reading photo data");
    String ReadFilter = "WHERE NAME='Mindy Martin'";
    Stream photoStream = client.Read_PHOTO(ReadFilter);
    Console.WriteLine("Photo data read -- writing to PhotoCopy.jpg");

    int count;
    int length = 0;
    byte[] buffer = new byte[4096];
    while ((count = photoStream.Read(buffer, 0, 4096)) > 0)
    {
        fs.Write(buffer, 0, count);
        length+=count;
    }
    Console.WriteLine("{0} bytes written to PhotoCopy.jpg", length);
}

Console.WriteLine("Photo updated and read back -- Hit <RETURN> to end");
Console.ReadLine();

```

10. Close the client as described in the snippet below:

```
client.Close();  
Console.WriteLine("Press any key to exit...");  
Console.ReadLine();
```

11. Build the project and then run it. The application updates the PHOTO column of the CUSTOMER table and then reads the content of the PHOTO column.

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Invoking Concurrent Programs in Oracle E-Business Suite Using the WCF Service Model

Oracle E-Business Suite exposes concurrent programs that you can execute to perform specific operations on Oracle applications. Each Oracle application has a set of standard concurrent programs (that are same across all operations) and certain concurrent programs that are specific to an Oracle application. The Oracle E-Business adapter exposes all concurrent programs as operations that adapter clients can invoke. For more information about how the adapter supports concurrent programs, see [Operations on Concurrent Programs](#).

Note

For the concurrent programs that do not expose their metadata, the Oracle E-Business adapter exposes 100 optional parameters for each of these concurrent programs. To invoke these concurrent programs successfully, the user must consult the Oracle E-Business Suite documentation to figure out the parameters for a concurrent program that require a value, and then specify them. An example of such a concurrent program is **Journal Import** (actual name: **GLLEZL**) in the **General Ledger** application.

About the Examples Used in this Topic

The example in this topic invokes the **MS_SAMPLE_COPY_EMP_DATA** concurrent program, followed by the **Get_Status** concurrent program to know the status of the first concurrent program. These concurrent programs are invoked from the **Application Object Library** application. The **MS_SAMPLE_COPY_EMP_DATA** is created by running the script provided with the samples. For more information about samples, see [Samples](#). A sample, **ConcurrentProgram_ServiceModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Client Class

The name of the WCF client generated for invoking the concurrent programs by the Oracle E-Business adapter is listed in the following table.

Artifact	WCF Client Name
Concurrent Program	ConcurrentPrograms_[APP_NAME]Client

[APP_NAME] = Actual name of the Oracle E-Business Suite application; for example, FND.

Method Signature for Invoking Concurrent Programs

The following table shows the method signature for the concurrent programs.

Operation	Method Signature
Concurrent program	public <return type> <Concurrent_program_name>(param 1, param 2, ...)

As an example, the following code shows the method signatures for a WCF client class generated for the **MS_SAMPLE_COPY_EMP_DATA** and **Get_Status** concurrent programs.

```
public partial class ConcurrentPrograms_FNDClient : System.ServiceModel.ClientBase<ConcurrentPrograms_FND>, ConcurrentPrograms_FND {

    public string MS_SAMPLE_COPY_EMP_DATA(schemas.microsoft.com.OracleEBS._2008._05.Options.SetOptions SetOptions,
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetPrintOptions SetPrintOptions,
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetRepeatOptions SetRepeatOptions,
        string Description, string StartTime);

    public bool GetStatusForConcurrentProgram(string RequestId, out string Phase, out string Status,
        out string DevPhase, out string DevStatus, out string Message);
}
```

In this snippet, **ConcurrentPrograms_FNDClient** is the name of the WCF class in the OracleEBSBindingClient.cs generated by the Add Adapter Service Reference Plug-in. **MS_SAMPLE_COPY_EMP_DATA** is the name of the method to invoke the

concurrent program. **GetStatusForConcurrentProgram** is the name of the method to invoke the concurrent program to get the status of the first concurrent program.

Note

GetStatusForConcurrentProgram is the actual name of the **Get_Status** concurrent program.

Creating a WCF Client to Invoke Concurrent Programs

The generic set of actions required to perform an operation on Oracle E-Business Suite using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **MS_SAMPLE_COPY_EMP_DATA** and **Get_Status** concurrent programs.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **MS_SAMPLE_COPY_EMP_DATA** and **Get_Status** concurrent programs. For more information about generating a WCF client class, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and add the following namespaces:

- `Microsoft.Adapters.OracleEBS`
- `System.ServiceModel`

5. Open the Program.cs file and create a client as described in the snippet below.

```
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs_instance_name");
ConcurrentPrograms_FNDClient client = new ConcurrentPrograms_FNDClient(binding, address);
```

In this snippet, `ConcurrentPrograms_FNDClient` is the WCF client defined in `OracleEBSBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in.

Note

In this snippet, you explicitly specify the binding and endpoint address in your application code. You can also use these values from the application configuration file, `app.config`, also generated by the Add Adapter Service Reference Plug-in. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

6. Set the credentials for the client.

```
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

7. Because you are invoking concurrent programs in an Oracle E-Business Suite application, you must set the application context. In this example, to set the application context, you specify the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

```
binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";
```

8. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

9. Invoke the **MS_SAMPLE_COPY_EMP_DATA** and **Get_Status** concurrent programs.

```
string RequestID;
bool Result;
string Phase;
string Status;
string DevPhase;
string DevStatus;
string Message;

try
{
    Console.WriteLine("Invoking the MS_SAMPLE_COPY_EMP_DATA concurrent program");
    RequestID = client.MS_SAMPLE_COPY_EMP_DATA(null, null, null, null, null);
    Console.WriteLine("The request ID generated for the concurrent program is : " + RequestID);
    Console.WriteLine("*****");
    Console.WriteLine("\nWaiting for 60 seconds for the concurrent program to be complete");
    System.Threading.Thread.Sleep(60000);
    Console.WriteLine("\nInvoking the Get_Status concurrent program");
    Result = client.GetStatusForConcurrentProgram(RequestID, out Phase, out Status, out DevPhase, out DevStatus, out Message);
    Console.WriteLine("\nResult is : " + Result);
    Console.WriteLine("Phase is : " + Phase);
    Console.WriteLine("Status is : " + Status);
    Console.WriteLine("DevPhase is : " + DevPhase);
    Console.WriteLine("DevStatus is : " + DevStatus);
    Console.WriteLine("Message is : " + Message);
    Console.WriteLine("*****");
    Console.WriteLine("\nHit <RETURN> to end");
    Console.ReadLine();
}
catch (Exception ex)
{
    Console.WriteLine("Exception : " + ex);
    throw;
}
```

10. Close the client as described in the snippet below:

```
client.Close();
```

11. Build the project and then run it. The application invokes the **MS_SAMPLE_COPY_EMP_DATA** and returns a request ID. The ID is then passed to the **Get_Status** concurrent program, which finally provides the status of the **MS_SAMPLE_COPY_EMP_DATA** column program.

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Invoking Request Sets in Oracle E-Business Suite Using the WCF Service Model

Microsoft BizTalk Adapter for Oracle E-Business Suite enables you to execute request sets in Oracle E-Business Suite. Request sets are divided into one or more stages, and each stage contains a set of reports and concurrent programs. For more information about how the adapter supports request sets, see [Operations on Request Sets](#).

The WCF Client Class

The name of the WCF client generated for invoking the request sets by the Oracle E-Business adapter is listed in the following table.

Artifact	WCF Client Name
Request Set	RequestSets_[APP_NAME]Client

[APP_NAME] = Actual name of the Oracle E-Business Suite application; for example, SQLAP.

Method Signature for Invoking Request Sets

The following table shows the method signature for request sets.

Operation	Method Signature
Request Set	public <return type> <request set name>(param 1, param 2, ...)

As an example, the following code shows the method signatures for a WCF client class generated for the **reqset_singlestage** request set.

Note
This is a custom request set and might not be available on your Oracle E-Business Solution instance.

```
public partial class RequestSets_SQLAPClient : System.ServiceModel.ClientBase<RequestSets_S
QLAP>, RequestSets_SQLAP{

    public string REQSTG(
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetRelClassOptions SetRelClassOptio
ns,
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetPrintOptions SetPrintOptions,
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetRepeatOptions SetRepeatOptions,
        schemas.microsoft.com.OracleEBS._2008._05.Options.SetNlsOptions SetNlsOptions,
        string StartTime,
        schemas.microsoft.com.OracleEBS._2008._05.RequestSetStage.SQLAP.REQSTG.set1 set1_set1
    );
}
```

In this snippet, **RequestSets_SQLAPClient** is the name of the WCF class in the OracleEBSBindingClient.cs generated by the Add Adapter Service Reference Plug-in. **REQSTG** is the name of the method to invoke the request set.

Creating a WCF Client to Invoke Request Sets

The generic set of actions required to perform an operation on Oracle E-Business Suite using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **reqset_singlestage** request set.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **reqset_singlestage** request set. For more information about generating a WCF client class, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and add the following namespaces:

- `Microsoft.Adapters.OracleEBS`
- `System.ServiceModel`

5. Open the Program.cs file and create a client as described in the snippet below.

```
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs_instance_name");
RequestSets_SQLAPClient client = new RequestSets_SQLAPClient(binding, address);
```

In this snippet, `RequestSets_SQLAPClient` is the WCF client defined in `OracleEBSBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in.

Note

In this snippet, you explicitly specify the binding and endpoint address in your application code. You can also use these values from the application configuration file, `app.config`, also generated by the Add Adapter Service Reference Plug-in. For more information about the different ways of specifying client binding, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

6. Set the credentials for the client.

```
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

7. Because you are invoking request sets in an Oracle E-Business Suite application, you must set the application context. In this example, to set the application context, you specify the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

```
binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";
```

8. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

9. Invoke the **reqset_singlestage** request set.

```

string RequestID;

schemas.microsoft.com.OracleEBS._2008._05.RequestSetStage.SQ LAP.REQSTG.set1 param =
    new schemas.microsoft.com.OracleEBS._2008._05.RequestSetStage.SQ LAP.REQSTG.set1();

param.INS PARAMPROG = new schemas.microsoft.com.OracleEBS._2008._05.RequestSetConcurren
tProgram.SQ LAP.REQSTG.set1.SQ LAP1.INS PARAMPROG();
param.INS PARAMPROG.p_id = "123";
param.INS PARAMPROG.p_name = "MyName";

try
{
    Console.WriteLine("Invoking the reqset_singlestage request set");
    RequestID = client.REQSTG(null, null, null, null, null, param);
    Console.WriteLine("The request ID generated for the request set is : " + RequestID
);
    Console.WriteLine("*****");
    Console.WriteLine("\nHit <RETURN> to end");
    Console.ReadLine();
}
catch (Exception ex)
{
    Console.WriteLine("Exception : " + ex);
    throw;
}

```

10. Close the client as described in the snippet below:

```
client.Close();
```

11. Build the project and then run it. The application invokes the **reqset_singlestage** request set and returns a request ID, which is written to the console.

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using WCF Service Model

The Oracle E-Business adapter exposes generic operations such as **ExecuteNonQuery**, **ExecuteReader**, and **ExecuteScalar**. You can use these operations to execute any statement on Oracle E-Business Suite. These operations differ based on the kind of response you get for the statement. For more information about how the adapter supports these operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

This topic demonstrates how to perform an **ExecuteReader** operation using the Oracle E-Business adapter using the WCF service model. You can follow the same set of procedures described in this topic to perform **ExecuteNonQuery** and **ExecuteScalar** operations.

About the Examples Used in this Topic

The example in this topic performs an **ExecuteReader** operation to perform a SELECT operation on the MS_SAMPLE_EMPLOYEE interface table. The table is created by running the script provided with the samples. For more information about samples, see [Samples](#). A sample, **ExecuteReader**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Client Class

The name of the WCF client generated for invoking generic operations (ExecuteNonQuery, ExecuteReader, or ExecuteScalar) using the Oracle E-Business adapter is listed in the following table.

Operations	WCF Client Name
ExecuteNonQuery, ExecuteReader, or ExecuteScalar	GenericOperation_Client

Method Signature for Invoking Generic Operations

The following table shows the signature for the method exposed to invoke the generic operations.

Operation	Method Signature
ExecuteNonQuery	int ExecuteNonQuery(string Query, string[] OutputRefCursorNames, out System.Data.DataSet[] OutputRefCursors)
ExecuteReader	System.Data.DataSet ExecuteReader(string Query)
ExecuteScalar	string ExecuteScalar(string Query)

As an example, the signature for the generic operation methods is shown in the following code snippet.

```
public partial class GenericOperation_Client : System.ServiceModel.ClientBase<GenericOperation_>, GenericOperation_ {
    public int ExecuteNonQuery(string Query, string[] OutputRefCursorNames, out System.Data.DataSet[] OutputRefCursors);
    public System.Data.DataSet ExecuteReader(string Query);
    public string ExecuteScalar(string Query);
}
```

In this snippet,

- `GenericOperation_Client` is the name of the class. This class is used to create a client to invoke the generic operation, `ExecuteReader`.
- `public System.Data.DataSet ExecuteReader(string Query)` is the method that you invoke to perform a SELECT statement on the MS_SAMPLE_EMPLOYEE interface table.

Creating a WCF Client to Invoke an ExecuteReader Operation

The generic set of actions required to perform an operation on Oracle E-Business Suite using a WCF client involves a set of

tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **ExecuteReader** operation.

To create a WCF client to invoke ExecuteReader operation

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **ExecuteReader** generic operation. This operation is available under the root node when you connect to the Oracle E-Business Suite using the Add Adapter Service Reference Plug-in. For more information about generating a WCF client class, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and add the following namespaces:

- `Microsoft.Adapters.OracleEBS`
- `System.ServiceModel`

5. In the Program.cs file, create a client as described in the snippet below.

```
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs-72-11");
GenericOperation_Client client = new GenericOperation_Client(binding, address);
```

In this snippet, `GenericOperation_Client` is the WCF client defined in `OracleEBSBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in.

Note

In this snippet, you explicitly specify the binding and endpoint address in your application code. You can use these values from the application configuration file, `app.config`, also generated by the Add Adapter Service Reference Plug-in. For more information about the different ways of specifying client binding, see [Specifying a Client Binding for the Oracle E-Business Suite](#).

6. Set the credentials for the client.

```
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

7. Because you are performing an operation on an interface table, you must set the application context. In this example, to set the application context, you specify the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

```
binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";
```

8. Open the client as described in the snippet below:

```
try
{
```

```

        Console.WriteLine("Opening Client...");
        client.Open();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception: " + ex.Message);
        throw;
    }
}

```

9. Invoke the **ExecuteReader** operation for performing the SELECT operation on MS_SAMPLE_EMPLOYEE table. Before you invoke the ExecuteReader operation, you must add the `System.Data` namespace to your code.

```

string query = "SELECT * FROM MS_SAMPLE_EMPLOYEE";
DataSet ds = client.ExecuteReader(query);

Console.WriteLine("Invoking the SELECT statement using ExecuteReader");
Console.WriteLine("*****");
foreach (DataTable tab in ds.Tables)
{
    foreach (DataRow row in tab.Rows)
    {
        Console.WriteLine("The details of the employee are: ");
        for (int i = 0; i < tab.Columns.Count; i++)
        {
            Console.WriteLine(row[i]);
        }
        Console.WriteLine();
    }
}
Console.WriteLine("*****");
Console.ReadLine();

```

10. Close the client as described in the snippet below:

```

client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();

```

11. Build the project and then run it. All the records in the MS_SAMPLE_EMPLOYEE table are displayed on the console.

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Polling Oracle E-Business Suite Using the WCF Service Model

You can configure the Oracle E-Business adapter to receive polling-based messages from the Oracle database. The adapter provides two ways of polling the Oracle database:

- **Using SELECT statements.** You can specify a simple SELECT statement to poll the Oracle database. The adapter executes the SELECT statement at specified intervals and returns the result to the adapter clients.
- **Using stored procedures.** You can specify a stored procedure to poll the Oracle database. The adapter executes the stored procedure at specified intervals and returns the result to the adapter clients.

The key difference in the two approaches is the way adapter clients specify a polling statement that the adapter uses to poll the Oracle database. While the polling statement for the first approach is a simple SELECT statement, the polling statement for the stored procedure approach is a request message that executes the stored procedure. Adapter clients specify the polling statement, for either approach, for the **PollingInput** binding property. For more information about the binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

The topics in this section provide instructions on how to poll using a SELECT statement and a stored procedure.

In This Section

- [Polling Using SELECT Statement Using the WCF Service Model](#)
- [Polling Using Stored Procedures Using the WCF Service Model](#)

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Polling Using SELECT Statement Using the WCF Service Model

You can configure the Oracle E-Business adapter to receive periodic data-change messages by using a SELECT statement to continuously poll the interface tables, interface views, tables and views in Oracle E-Business Suite. You can specify a SELECT statement as a polling statement that the adapter executes periodically to poll Oracle E-Business Suite. You can also specify a post-poll PL/SQL code block that the adapter executes after the polling statement is executed.

To enable polling, you must specify certain binding properties as described in this topic. For more information about how the adapter supports polling, see [Support for Inbound Calls Using Polling](#).

Configuring a Polling Operation with Oracle E-Business Suite Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure the adapter to receive data change messages. You must specify these binding properties while running the polling application.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the SELECT statement you specify for the PollingInput binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle E-Business adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingInput	Specifies the polling statement. To poll using a SELECT statement, you must specify a SELECT statement for this binding property. The default is null. You must specify a value for PollingInput binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action from the service interface generated for the operation using the Add Adapter Service Reference Visual Studio Plug-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed.
PollWhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to poll the Oracle database, read further.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving data change messages using SELECT statements, you poll the **MS_SAMPLE_EMPLOYEE** interface table in the **Application Object Library** application. This table is created when you run the create_apps_artifacts.sql script provided with the samples to create these objects in Oracle E-Business Suite.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the interface table being polled (MS_SAMPLE_EMPLOYEE) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE
```

This ensures that the adapter executes the polling statement only when the MS_SAMPLE_EMPLOYEE interface table has some records.

- Specify a SELECT statement for the **PollingInput** binding property. This statement retrieves all the rows in the MS_SAMPLE_EMPLOYEE interface table. In this example, you can set this binding property as:

```
SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE
```

Note

For information about the FOR UPDATE clause used in the SELECT statement, see [Specifying a FOR UPDATE Clause in the Polling Statement](#).

- Specify a DELETE statement as part of the **PostPollStatement** binding property. This statement will delete all data from MS_SAMPLE_EMPLOYEE interface table. In this example, you can set this binding property as:

```
DELETE FROM MS_SAMPLE_EMPLOYEE
```

After this happens, the next time the statement specified for **PollingInput** will be executed, it will not fetch any data.

- Until more data is added to the MS_SAMPLE_EMPLOYEE interface table, you will not get any polling messages so you must repopulate the MS_SAMPLE_EMPLOYEE interface table with new records. You can do so by running the insert_apps_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

Configuring Polling in the WCF Service Model

To poll an interface table using the Oracle E-Business adapter with the WCF service model, you must:

- Generate a WCF service contract (interface) for the **Poll** operation on the MS_SAMPLE_EMPLOYEE interface table. To do this, you could use the Add Adapter Service Reference Plug-in.
- Implement a WCF service from this interface.
- Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic poll the MS_SAMPLE_EMPLOYEE interface table. A script to generate the table is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **SelectPolling_ServiceModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **Polling** operation. For more information about generating a WCF service contract, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **Poll** operation on MS_SAMPLE_EMPLOYEE

interface table.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/Oracl
eEBS/", ConfigurationName="InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE")]
public interface InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.mic
rosoft.com/Oracl
eEBS/2008/05/InterfaceTables/FND/APPS/MS_SAMPLE_EMPLOYEE) of message Poll
// does not match the default value (http://schemas.microsoft.com/Oracl
eEBS/)
[System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="InterfaceTables/
Poll/FND/APPS/MS_SAMPLE_EMPLOYEE")]
    void Poll(Poll request);
}
```

The Message Contracts

Following is the message contract for the **Poll** operation on MS_SAMPLE_EMPLOYEE interface table.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="Poll", WrapperNamespace="http://
schemas.microsoft.com/Oracl
eEBS/2008/05/InterfaceTables/FND/APPS/MS_SAMPLE" +
    "_EMPLOYEE", IsWrapped=true)]
public partial class Poll {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com
/Oracl
eEBS/2008/05/InterfaceTables/FND/APPS/MS_SAMPLE" +
        "_EMPLOYEE", Order=0)]
    public schemas.microsoft.com.Oracl
eEBS._2008._05.TableViewRecord.APPS.MS_SAMPLE_EMPLOYE
E.SelectRecord[] DATA;

    public Poll() {
    }

    public Poll(schemas.microsoft.com.Oracl
eEBS._2008._05.TableViewRecord.APPS.MS_SAMPLE_EM
PLOYEE.SelectRecord[] DATA) {
        this.DATA = DATA;
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is OracleEBSBindingService.cs. You can insert the logic to process the **Poll** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```
namespace OracleEBSBindingNamespace {

    public class OracleEBSBindingService : InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE {

        // CODEGEN: Generating message contract since the wrapper namespace (http://schemas
.microsoft.com/Oracl
eEBS/2008/05/InterfaceTables/FND/APPS/MS_SAMPLE_EMPLOYEE) of message Po
ll
// does not match the default value (http://schemas.microsoft.com/Oracl
eEBS/)
        public virtual void Poll(Poll request) {
            throw new System.NotImplementedException("The method or operation is not implem
ented.");
        }
    }
}
```

Receiving Inbound Messages for the Poll Operation Using a SELECT Statement

This section provides instructions on how to write a .NET application to receive inbound polling messages using the Oracle E-Business adapter.

To receive polling messages using a SELECT statement

1. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **Poll** operation on the MS_SAMPLE_EMPLOYEE interface table. For more information, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.
2. Implement a WCF service from the interface and helper classes generated in step 1. The **Poll** method of this class can throw an exception to abort the polling transaction, if an error is encountered processing the data received from the **Poll** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **Poll** method, you can implement your application logic directly. This class can be found in OracleEBSBindingService.cs. This code in this example sub-classes the **OracleEBSBindingService** class. In this code, the polling message received and is written to the console.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class PollingService : OracleEBSBindingNamespace.OracleEBSBindingService
{
    public override void Poll(Poll request)
    {
        Console.WriteLine("\nNew Polling Records Received");
        Console.WriteLine("*****");
        Console.WriteLine("Emp No\tName\tDesignation\tSalary");
        for (int i = 0; i < request.DATA.Length; i++)
        {
            Console.WriteLine("{0}\t{1}\t{2}\t{3}",
                request.DATA[i].EMP_NO,
                request.DATA[i].NAME,
                request.DATA[i].DESIGNATION,
                request.DATA[i].SALARY);
        }
        Console.WriteLine("*****");
        Console.WriteLine("\nHit <RETURN> to stop polling");
    }
}
```

3. You must implement the following class to avoid passing credentials as part of the URI. In the latter part of the application, you will instantiate this class to pass on the credentials.

```
class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }
}
```

```

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new PollingCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

```

4. Create an **OracleEBSBinding** and configure the polling operation by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType**, **PolledDataAvailableStatement**, **PollingInput**, and **PollingAction** binding properties.

```

OracleEBSBinding binding = new OracleEBSBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE";
binding.PollingInput = "SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE";
binding.PollingAction = "InterfaceTables/Pol/FND/APPS/MS_SAMPLE_EMPLOYEE";
binding.PostPollStatement = "DELETE FROM MS_SAMPLE_EMPLOYEE";

```

5. Because you are polling an interface table, you must also set the applications context. For more information about application context and the binding properties required for setting application context, see [Setting Application Context](#).

```

binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";

```

6. Specify Oracle E-Business Suite credentials by instantiating the **PollingCredentials** class you created in Step 3.

```

PollingCredentials credentials = new PollingCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

```

7. Create an instance of the WCF service created in step 2.

```

// create service instance
PollingService service = new PollingService();

```

8. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. The base connection URI cannot contain the inbound ID, if specified. You must also pass the credentials here.

```

// Enable service host
Uri[] baseUri = new Uri[] { new Uri("oracleebs://ebs_instance_name") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);

```

9. Add a service endpoint to the service host. To do this:

- Use the binding created in step 4.
- Specify a connection URI that contains credentials and, if required, an inbound ID.

- Specify the contract as "InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE" to poll the MS_SAMPLE_EMPLOYEE interface table.

```
// Add service endpoint: be sure to specify InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE as the contract
Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name");
serviceHost.AddServiceEndpoint("InterfaceTables_FND_APPS_MS_SAMPLE_EMPLOYEE", binding, ConnectionUri);
```

10. To receive polling data, open the service host. The adapter will return data whenever the query returns a result set.

```
// Open the service host to begin polling
serviceHost.Open();
```

11. To terminate polling, close the service host.

◆ Important

The adapter will continue to poll until the service host is closed.

```
serviceHost.Close();
```

Example

The following example shows a polling application that polls the MS_SAMPLE_EMPLOYEE interface table. The **PollingInput** property contains the select statement that reads all the data from the MS_SAMPLE_EMPLOYEE table and the post poll statement deletes all the data from the same table. The first polling message gives all the records from the MS_SAMPLE_EMPLOYEE interface table. Subsequent polling messages will not contain any records because the post poll statement deletes the records. Until more data is added to the MS_SAMPLE_EMPLOYEE interface table, you will not get any polling messages. So, you must repopulate the MS_SAMPLE_EMPLOYEE interface table with new records. You can do so by running the insert_apps_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table. The adapter will continue to poll until you close the service host by pressing <RETURN>.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Adapters.OracleEBS;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace SelectPolling_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class PollingService : OracleEBSBindingNamespace.OracleEBSBindingService
    {
        public override void Poll(Poll request)
        {
            Console.WriteLine("\nNew Polling Records Received");
            Console.WriteLine("*****");
            Console.WriteLine("Emp No\tName\tDesignation\tSalary");
            for (int i = 0; i < request.DATA.Length; i++)
            {
                Console.WriteLine("{0}\t{1}\t{2}\t{3}",
                    request.DATA[i].EMP_NO,
```

```

        request.DATA[i].NAME,
        request.DATA[i].DESIGNATION,
        request.DATA[i].SALARY);
    }
    Console.WriteLine("*****");
    Console.WriteLine("\nHit <RETURN> to stop polling");
}
}

class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
dingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
tBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
HostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new PollingCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

class Program
{
    static void Main(string[] args)
    {
        ServiceHost serviceHost = null;
        try
        {
            Console.WriteLine("Sample started...");
            Console.WriteLine("Press any key to start polling...");
            Console.ReadLine();

            OracleEBSBinding binding = new OracleEBSBinding();
            binding.InboundOperationType = InboundOperation.Polling;
            binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM MS_SAMPLE_EMP
LOYEE";

            binding.PollingInput = "SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE";
            binding.PollingAction = "InterfaceTables/Poll/FND/APPS/MS_SAMPLE_EMPLOYEE";
            binding.PostPollStatement = "DELETE FROM MS_SAMPLE_EMPLOYEE";
            binding.OracleUserName = "myOracleEBSUserName";
            binding.OraclePassword = "myOracleEBSPassword";
            binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";

            // This URI is used to specify the address for the ServiceEndpoint
            // It must contain the InboundId that was used to generate
            // the WCF service callback interface
            Uri ConnectionUri = new Uri("oracleEBS://ebs_instance_name");

            // This URI is used to initialize the ServiceHost. It cannot contain
            // an InboundID; otherwise, an exception is thrown when
            // the ServiceHost is initialized.
            Uri[] baseUri = new Uri[] { new Uri("oracleEBS://ebs_instance_name") };

            PollingCredentials credentials = new PollingCredentials();

```


Polling Using Stored Procedures Using the WCF Service Model

You can configure the Oracle E-Business adapter to receive periodic data-change messages by using stored procedures to periodically poll the Oracle database. You can specify a stored procedure as a polling statement that the adapter executes periodically to poll the Oracle database.

To enable polling, you must specify certain binding properties as described in this topic. For more information about how the adapter supports polling, see [Support for Inbound Calls Using Polling](#).

Configuring a Polling Operation with Oracle E-Business Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure the adapter to receive data-change messages. You must specify these binding properties while running the polling application.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform a Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the stored procedure you specified for the PollingInput binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle E-Business adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingInput	Specifies the polling statement. To poll using a stored procedure, you must specify the entire request message for this binding property. The request message must be the same that you send to the adapter for invoking the stored procedure as an outbound operation. The default is null. You must specify a value for PollingInput binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action from the service interface generated for the operation using the Add Adapter Service Reference Visual Studio Plug-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed.
PollWhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to poll, read the following sections.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving data change messages using stored procedures, you use the GET_ACTIVITYS stored procedure to poll the ACCOUNTACTIVITY table in the Oracle database. This stored procedure is available with the ACCOUNT_PKG package. You can run the SQL scripts provided with the samples to create these objects in the database.

Note

The example in this topic polls the ACCOUNTACTIVITY table, which is a base database table created by running the scripts provided with the samples. You must perform similar procedures as described in this topic to poll any other table, including interface tables.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the table being polled (ACCOUNTACTIVITY) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM ACCOUNTACTIVITY
```

This ensures that the adapter executes the polling statement only when the ACCOUNTACTIVITY table has some records.

- Execute a stored procedure, GET_ACTIVITYS, by providing the request message as part of the **PollingInput** binding property. This stored procedure will retrieve all the rows in the ACCOUNTACTIVITY table and you will get a response message from the adapter.
- Execute a PL/SQL block as part of the **PostPollStatement** binding property. This statement will move all data from ACCOUNTACTIVITY table to another table in the database. After this happens, the next time **PollingInput** is executed, it will not fetch any data and hence the GET_ACTIVITYS stored procedure will return an empty response message.
- Until more data is added to the ACCOUNTACTIVITY table, you will continue to get empty response messages so you must repopulate the ACCOUNTACTIVITY table with new records. You can do so by running the more_activity_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

Configuring Polling in the WCF Service Model

To poll using stored procedures with the Oracle E-Business adapter with the WCF service model, you must:

- Generate a WCF service contract (interface) for the stored procedure using which you are going to poll. For this example, you must generate the WCF service contract for the **GET_ACTIVITYS** stored procedure as an inbound operation. To do this, you could use the Add Adapter Service Reference Plug-in.
- Implement a WCF service from this interface.
- Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic poll the ACCOUNTACTIVITY database table using the GET_ACTIVITYS stored procedure. A script to generate the table and the stored procedure is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **StoredProcPolling_ServiceModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **GET_ACTIVITYS** inbound operation. For more information about generating a WCF service contract, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **GET_ACTIVITYS** inbound operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]  
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/", ConfigurationName="PollingPackageApis_APPS_ACCOUNT_PKG")]
```

```

public interface PollingPackageApis_APPS_ACCOUNT_PKG {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/OracleEBS/2008/05/PollingPackageApis/APPs/ACCOUNT_PKG) of message GET_ACTIVITYS
    // does not match the default value (http://schemas.microsoft.com/OracleEBS/)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="PollingPackageApis/APPs/ACCOUNT_PKG/GET_ACTIVITYS")]
    void GET_ACTIVITYS(GET_ACTIVITYS request);
}

```

The Message Contracts

Following is the message contract for the **GET_ACTIVITYS** inbound operation.

```

[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="GET_ACTIVITYS", WrapperNamespace="http://schemas.microsoft.com/OracleEBS/2008/05/PollingPackageApis/APPs/ACCOUNT_PK" + "G", IsWrapped=true)]
public partial class GET_ACTIVITYS {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/PollingPackageApis/APPs/ACCOUNT_PK" + "G", Order=0)]
    public schemas.microsoft.com.OracleEBS._2008._05.RecordTypes.APPs.ACCOUNT_PKG.GET_ACTIVITYS.OUTRECSRecord[] OUTRECS;

    public GET_ACTIVITYS() {
    }

    public GET_ACTIVITYS(schemas.microsoft.com.OracleEBS._2008._05.RecordTypes.APPs.ACCOUNT_PKG.GET_ACTIVITYS.OUTRECSRecord[] OUTRECS) {
        this.OUTRECS = OUTRECS;
    }
}

```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is OracleEBSBindingService.cs. You can insert the logic to process the **GET_ACTIVITYS** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```

namespace OracleEBSBindingNamespace {

    public class OracleEBSBindingService : PollingPackageApis_APPS_ACCOUNT_PKG {

        // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/OracleEBS/2008/05/PollingPackageApis/APPs/ACCOUNT_PKG) of message GET_ACTIVITYS
        // does not match the default value (http://schemas.microsoft.com/OracleEBS/)
        public virtual void GET_ACTIVITYS(GET_ACTIVITYS request) {
            throw new System.NotImplementedException("The method or operation is not implemented.");
        }
    }
}

```

Receiving Inbound Messages For Polling Using a Stored Procedure

This section provides instructions on how to write a .NET application to receive inbound polling messages using the Oracle E-Business adapter.

To receive polling messages using a stored procedure

1. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the

GET_ACTIVITYS inbound operation. For more information, see

[Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

2. Implement a WCF service from the interface and helper classes generated in step 1. The **GET_ACTIVITYS** method of this class can throw an exception to abort the polling transaction, if an error is encountered processing the data received from the **GET_ACTIVITYS** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **GET_ACTIVITYS** method, you can implement your application logic directly. This class can be found in OracleEBSBindingService.cs. This code in this example sub-classes the **OracleEBSBindingService** class. In this code, the polling message received and is written to the console.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class PollingService : OracleEBSBindingNamespace.OracleEBSBindingService
{
    public override void GET_ACTIVITYS(GET_ACTIVITYS request)
    {
        Console.WriteLine("\nNew Polling Records Received");
        Console.WriteLine("*****");
        Console.WriteLine("Tx Id\tAccount\tAmount\tProcessed");
        for (int i = 0; i < request.OUTRECS.Length; i++)
        {
            Console.WriteLine("{0}\t{1}\t{2}\t{3}",
                request.OUTRECS[i].TID,
                request.OUTRECS[i].ACCOUNT,
                request.OUTRECS[i].AMOUNT,
                request.OUTRECS[i].PROCESSED);
        }
        Console.WriteLine("*****");
        Console.WriteLine("\nHit <RETURN> to stop polling");
    }
}
```

3. You must implement the following class to avoid passing credentials as part of the URI. In the latter part of the application, you will instantiate this class to pass on the credentials.

```
class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }
}
```

```

protected override ClientCredentials CloneCore()
{
    ClientCredentials clone = new PollingCredentials();
    clone.UserName.UserName = this.UserName.UserName;
    clone.UserName.Password = this.UserName.Password;
    return clone;
}
}

```

4. Create an **OracleEBSBinding** and configure the polling operation by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType**, **PolledDataAvailableStatement**, **PollingInput**, and **PollingAction** binding properties.

```

OracleEBSBinding binding = new OracleEBSBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM ACCOUNTACTIVITY";
binding.PollingInput = "<GET_ACTIVITYYS xmlns='http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/APPS/ACCOUNT_PKG'><INRECS>OPEN ? FOR SELECT * FROM ACCOUNTACTIVITY</INRECS></GET_ACTIVITYYS>";
binding.PollingAction = "PollingPackageApis/APPS/ACCOUNT_PKG/GET_ACTIVITYYS";
binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";

```

Note

Note that the value for the **PollingInput** binding property contains the request message for invoking the **GET_ACTIVITYYS** stored procedure as an outbound operation.

Important

In this example, because you are polling a database table, you do not need to set the applications context. However, if you were polling an interface table, you must set the applications context by specifying the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

5. Specify Oracle E-Business Suite credentials by instantiating the **PollingCredentials** class you created in Step 3.

```

PollingCredentials credentials = new PollingCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

```

6. Create an instance of the WCF service created in step 2.

```

// create service instance
PollingService service = new PollingService();

```

7. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. The base connection URI cannot contain the inbound ID, if specified. You must also pass the credentials here.

```

// Enable service host
Uri[] baseUri = new Uri[] { new Uri("oracleEBS://ebs_instance_name") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);

```

8. Add a service endpoint to the service host. To do this:

- Use the binding created in step 4.

- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "PollingPackageApis_APPS_ACCOUNT_PKG" to poll the MS_SAMPLE_EMPLOYEE interface table.

```
// Add service endpoint: be sure to specify PollingPackageApis_APPS_ACCOUNT_PKG as the
contract
Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name");
serviceHost.AddServiceEndpoint("PollingPackageApis_APPS_ACCOUNT_PKG", binding, Connect
ionUri);
```

9. To receive polling data, open the service host. The adapter will return data whenever the query returns a result set.

```
// Open the service host to begin polling
serviceHost.Open();
```

10. To terminate polling, close the service host.

◆ Important

The adapter will continue to poll until the service host is closed.

```
serviceHost.Close();
```

Example

The following example shows a polling application that polls the ACCOUNTACTIVITY database table using the GET_ACTIVITYS stored procedure. The **PollingInput** binding property contains the request message to invoke the GET_ACTIVITYS stored procedure that reads all the data from the ACCOUNTACTIVITY table and the post poll statement moves all the data from ACCOUNTACTIVITY to ACTIVITYHISTORY table.

The first polling message gives all the records from the ACCOUNTACTIVITY table. Subsequent polling messages will not contain any records because the post poll statement deletes the records. Until more data is added to the ACCOUNTACTIVITY table, you will not get any polling messages so you must repopulate the ACCOUNTACTIVITY table with new records. You can do so by running the more_activity_data.sql script provided with the samples.

After you run this script, the next polling operation will fetch the new records inserted into the table. The adapter will continue to poll until you close the service host by pressing <RETURN>.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Microsoft.Adapters.OracleEBS;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace StoredProcPolling_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class PollingService : OracleEBSBindingNamespace.OracleEBSBindingService
    {
        public override void GET_ACTIVITYS(GET_ACTIVITYS request)
        {
            Console.WriteLine("\nNew Polling Records Received");
        }
    }
}
```

```

Console.WriteLine("*****");
Console.WriteLine("Tx Id\tAccount\tAmount\tProcessed");
for (int i = 0; i < request.OUTRECS.Length; i++)
{
    Console.WriteLine("{0}\t{1}\t{2}\t{3}",
        request.OUTRECS[i].TID,
        request.OUTRECS[i].ACCOUNT,
        request.OUTRECS[i].AMOUNT,
        request.OUTRECS[i].PROCESSED);
}
Console.WriteLine("*****");
Console.WriteLine("\nHit <RETURN> to stop polling");
}
}

class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
dingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
tBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
HostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new PollingCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

class Program
{
    static void Main(string[] args)
    {
        ServiceHost serviceHost = null;
        try
        {
            Console.WriteLine("Sample started...");
            Console.WriteLine("Press any key to start polling...");
            Console.ReadLine();

            OracleEBSBinding binding = new OracleEBSBinding();
            binding.InboundOperationType = InboundOperation.Polling;
            binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM ACCOUNTACTIVI
TY";
            binding.PollingInput = "<GET_ACTIVITYS xmlns='http://schemas.microsoft.com/
OracleEBS/2008/05/PackageApis/APPS/ACCOUNT_PKG'><INRECS>OPEN ? FOR SELECT * FROM ACCOUNTACT
IVITY</INRECS></GET_ACTIVITYS>";
            binding.PollingAction = "PollingPackageApis/APPS/ACCOUNT_PKG/GET_ACTIVITYS"
;
            binding.PostPollStatement = "BEGIN ACCOUNT_PKG.PROCESS_ACTIVITY(); END;";

            // This URI is used to specify the address for the ServiceEndpoint
            // It must contain the InboundId that was used to generate
            // the WCF service callback interface
            Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name");

```


Receiving Database Change Notifications Using the WCF Service Model

This topic demonstrates how to configure the Oracle E-Business adapter to receive query notification messages from an Oracle database. To demonstrate notifications, consider a table, ACCOUNTACTIVITY, with a "Processed" column. When a new record is inserted to this table, the value of the Status column is set to "n." You can configure the adapter to receive notifications by registering for notifications using a SQL statement that retrieves all records that have "Processed" column as "n." You can do so by specifying the SQL statement for the **NotificationStatement** binding property. After the adapter client receives the notification, it can contain the logic to do any subsequent tasks on the Oracle database. In this example, for the sake of simplicity, the adapter client lists all the records in the table that have the "Processed" column as "n."

Configuring Notifications with the Oracle E-Business Adapter Binding Properties

The table below summarizes the Oracle E-Business adapter binding properties that you use to configure receiving notifications from Oracle E-Business Suite. You must specify these binding properties while running the .NET application to receive notifications.

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationPort	Specifies the port number that ODP.NET must open to listen for database change notification from Oracle database.
NotificationStatement	Specifies the Select statement used to register for query notifications. The adapter gets a notification message only when the result set for the specified Select statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to receive notifications from Oracle E-Business Suite, read the remainder of this topic.

Configuring Notifications Using the WCF Service Model

To receive the notifications using the WCF service model, you must:

- Generate a WCF service contract (interface) for the **Notification** operation from the metadata exposed by the adapter. To do this, you could use the Add Adapter Service Reference Plug-in.
- Generate a WCF client for the **Select** operation on the ACCOUNTACTIVITY table. To do this, you could use the Add Adapter Service Reference Plug-in.
- Implement a WCF service from this interface.
- Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic receives notification for the ACCOUNTACTIVITY table. A script to generate the table is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **Notification_ServiceModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **Notification** operation. For more information about generating a WCF service contract, see

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **Notification** operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/", ConfigurationName="Notification_")]
public interface Notification_ {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/OracleEBS/2008/05/Notification/) of message Notification
    // does not match the default value (http://schemas.microsoft.com/OracleEBS/)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="Notification")]
    void Notification(Notification request);
}
```

The Message Contracts

Following is the message contract for the Notification operation.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="Notification", WrapperNamespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", IsWrapped=true)]
public partial class Notification {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", Order=0)]
    public schemas.microsoft.com.OracleEBS._2008._05.Notification.NotificationDetails[] Details;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", Order=1)]
    public string Info;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", Order=2)]
    public string[] ResourceNames;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", Order=3)]
    public string Source;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/OracleEBS/2008/05/Notification/", Order=4)]
    public string Type;

    public Notification() {
    }

    public Notification(schemas.microsoft.com.OracleEBS._2008._05.Notification.NotificationDetails[] Details, string Info, string[] ResourceNames, string Source, string Type) {
        this.Details = Details;
        this.Info = Info;
        this.ResourceNames = ResourceNames;
        this.Source = Source;
        this.Type = Type;
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is OracleEBSBindingService.cs. You can insert the logic to process the **Notification** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter

```

namespace OracleEBSBindingNamespace {

    public class OracleEBSBindingService : Notification_ {

        // CODEGEN: Generating message contract since the wrapper namespace (http://schemas
        .microsoft.com/OracleEBS/2008/05/Notification/) of message Notification
        // does not match the default value (http://schemas.microsoft.com/OracleEBS/)
        public virtual void Notification(Notification request) {
            throw new System.NotImplementedException("The method or operation is not implem
            ented.");
        }
    }
}

```

Receiving Query Notifications Using WCF Service Model

This section provides instructions on how to write a .NET application to receive query notifications using the Oracle E-Business adapter.

To receive query notifications

1. Use the Add Adapter Service Reference Plug-in to generate a WCF client for **Select** operation on the **ACCOUNTACTIVITY** table. You will use this client to perform Select operations after receiving a notification message. Add a new class, TableOperation.cs, to your project and add the following code snippet to perform a Select operation.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Notification_ServiceModel
{
    class TableOperation
    {
        public void TableOp()
        {
            ////////////////////////////////////////////////////////////////////
            // CREATING THE CLIENT AND SETTING CLIENT CREDENTIALS
            ////////////////////////////////////////////////////////////////////

            Tables_APPS_ACCOUNTACTIVITYClient client = new Tables_APPS_ACCOUNTACTIVITY
            Client();

            client.ClientCredentials.UserName.UserName = "<Enter user name here>";
            client.ClientCredentials.UserName.Password = "<Enter password here>";

            ////////////////////////////////////////////////////////////////////
            // OPENING THE CLIENT
            ////////////////////////////////////////////////////////////////////
            try
            {
                Console.WriteLine("Opening the client ...");
                client.Open();
            }
            catch (Exception ex)
            {
                Console.WriteLine("Exception: " + ex.Message);
                throw;
            }
        }
    }
}

```

```

////////////////////////////////////
////////////////////////////////////
// SELECTING THE LAST INSERTED VALUES
////////////////////////////////////
////////////////////////////////////
Console.WriteLine("The application will now select the last inserted recor
d");

        schemas.microsoft.com.OracleEBS._2008._05.TableViewRecord.APPS.ACCOUNTACTI
VITY.SelectRecord[] selectRecords;

        try
        {
            selectRecords = client.Select("", "WHERE PROCESSED = 'n'");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Exception: " + ex.Message);
            throw;
        }

        Console.WriteLine("The details of the newly added records are:");
        Console.WriteLine("*****");
        for (int i = 0; i < selectRecords.Length; i++)
        {
            Console.WriteLine("Transaction ID   : " + selectRecords[i].TID);
            Console.WriteLine("Account ID       : " + selectRecords[i].ACCOUNT);
            Console.WriteLine("Processed Status : " + selectRecords[i].PROCESSED);
            Console.WriteLine();
        }
        Console.WriteLine("*****");
    }
}
}
}

```

2. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **Notification** operation.

For more information, see [Generating a WCF Client or a WCF Service Contract for Oracle E-Business Solution Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

3. Implement a WCF service from the interface and helper classes generated in step 2. The **Notification** method of this class can throw an exception to abort the operation, if an error is encountered processing the data received from the **Notification** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **Notification** method, you can implement your application logic directly. This class can be found in OracleEBSBindingService.cs. This code in this example sub-classes the **OracleEBSBindingService** class. In this code, the notification message received is written to the console. Additionally, the **TableOp** method within the **TableOperation** class is invoked to perform the Select operation.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class NotificationService : OracleEBSBindingNamespace.OracleEBSBindingService
{

```

```

public override void Notification(Notification request)
{
    Console.WriteLine("\nNew Notification Received");
    Console.WriteLine("*****");
    Console.WriteLine(request.Info);
    Console.WriteLine(request.Source);
    Console.WriteLine(request.Type);
    Console.WriteLine("*****");

    TableOperation Ops = new TableOperation();
    Ops.TableOp();
}
}

```

4. You must implement the following class to pass credentials for the Oracle E-Business Suite. In the latter part of the application, you will instantiate this class to pass on the credentials.

```

class NotificationCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new NotificationCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

```

5. Create an **OracleEBSBinding** and configure the adapter to receive query notifications by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType** and **NotificationStatement** binding properties.

```

OracleEBSBinding binding = new OracleEBSBinding();
binding.InboundOperationType = InboundOperation.Notification;
binding.NotificationStatement = "SELECT TID,ACCOUNT,PROCESSED FROM APPS.ACCOUNTACTIVITY WHERE PROCESSED = 'n'";
binding.NotifyOnListenerStart = true;
binding.NotificationPort = 10;

```

◆ Important

The value for the **NotificationPort** binding property must be set to the same port number that you must have added to the Windows Firewall exceptions list. For instructions on how to add ports to Windows Firewall exceptions list, see <http://go.microsoft.com/fwlink/?LinkID=196959>.

◆ Important

If you do not set the **NotificationPort** binding property, the adapter will assume the default value of -1 for this binding property. In such a case, you will have to completely disable Windows Firewall to receive notification messages.

- Specify Oracle E-Business Suite credentials by instantiating the **NotificationCredentials** class you created in Step 4.

```
NotificationCredentials credentials = new NotificationCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";
```

- Create an instance of the WCF service created in step 3.

```
// create service instance
NotificationService service = new NotificationService();
```

- Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. You must also specify the credentials here.

```
// Enable service host
Uri[] baseUri = new Uri[] { new Uri("oracleebss://ebs_instance_name") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);
```

- Add a service endpoint to the service host. To do this:

- Use the binding created in step 5.
- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "Notification_".

```
// Add service endpoint: be sure to specify Notification_ as the contract
Uri ConnectionUri = new Uri("oracleebss://ebs_instance_name?");
serviceHost.AddServiceEndpoint("Notification_", binding, ConnectionUri);
```

- To receive notification message, open the service host.

```
// Open the service host to begin receiving notifications
serviceHost.Open();
```

- To stop receiving notifications, close the service host.

```
serviceHost.Close();
```

Example

The following example shows a .NET application to receive notification messages for the ACCOUNTACTIVITY table.

Note

The following code snippet instantiates a **TableOperation.cs** class and invokes the **TableOp** method. The class and the method are described in Step 1.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Microsoft.Adapters.OracleEBS;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace Notification_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class NotificationService : OracleEBSBindingNamespace.OracleEBSBindingService
    {
        public override void Notification(Notification request)
        {
            Console.WriteLine("\nNew Notification Received");
            Console.WriteLine("*****");
            Console.WriteLine(request.Info);
            Console.WriteLine(request.Source);
            Console.WriteLine(request.Type);
            Console.WriteLine("*****");

            TableOperation Ops = new TableOperation();
            Ops.TableOp();
        }
    }

    class NotificationCredentials : ClientCredentials, IServiceBehavior
    {
        public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
        Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
        dingParameters)
        {
            bindingParameters.Add(this);
        }

        public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
        tBase serviceHostBase)
        { }

        public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
        HostBase)
        { }

        protected override ClientCredentials CloneCore()
        {
            ClientCredentials clone = new NotificationCredentials();
            clone.UserName.UserName = this.UserName.UserName;
            clone.UserName.Password = this.UserName.Password;
            return clone;
        }
    }

    class Program
    {
```

```

static void Main(string[] args)
{
    ServiceHost serviceHost = null;
    try
    {
        Console.WriteLine("Sample started...");
        Console.WriteLine("Press any key to start receiving notifications...");
        Console.ReadLine();

        OracleEBSBinding binding = new OracleEBSBinding();
        binding.InboundOperationType = InboundOperation.Notification;
        binding.NotificationStatement = "SELECT TID,ACCOUNT,PROCESSED FROM APPS.ACCO
OUNTACTIVITY WHERE PROCESSED = 'n'";
        binding.NotifyOnListenerStart = true;
        binding.NotificationPort = 10;

        // This URI is used to specify the address for the ServiceEndpoint
        // It must contain the InboundId that was used to generate
        // the WCF service callback interface
        Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name");

        // This URI is used to initialize the ServiceHost. It cannot contain
        // an InboundID; otherwise,an exception is thrown when
        // the ServiceHost is initialized.
        Uri[] baseUri = new Uri[] { new Uri("oracleebs://ebs_instance_name") };

        NotificationCredentials credentials = new NotificationCredentials();
        credentials.UserName.UserName = "<Enter user name here>";
        credentials.UserName.Password = "<Enter password here>";

        Console.WriteLine("Opening service host...");
        NotificationService service = new NotificationService();
        serviceHost = new ServiceHost(service, baseUri);
        serviceHost.Description.Behaviors.Add(credentials);
        serviceHost.AddServiceEndpoint("Notification_", binding, ConnectionUri);
        serviceHost.Open();
        Console.WriteLine("Service host opened...");
        Console.WriteLine("Waiting for notification...");

        Console.WriteLine("\nHit <RETURN> to stop receiving notification");
        Console.ReadLine();
    }
    catch (Exception e)
    {
        Console.WriteLine("Exception :" + e.Message);
        Console.ReadLine();

        /* If there is an error it will be specified in the inner exception */
        if (e.InnerException != null)
        {
            Console.WriteLine("InnerException: " + e.InnerException.Message);
            Console.ReadLine();
        }
    }
    finally
    {
        // IMPORTANT: you must close the ServiceHost to stop polling
        if (serviceHost.State == CommunicationState.Opened)
            serviceHost.Close();
        else
            serviceHost.Abort();
    }
}
}
}
}

```

See Also

Other Resources

[Developing Applications By Using the WCF Service Model](#)

Developing Applications by Using the WCF Channel Model

You can use the Windows Communication Foundation (WCF) channel model to consume the Microsoft BizTalk Adapter for Oracle E-Business Suite by sending XML messages directly over a channel instance created with the Oracle EBS Binding.

One advantage of using the WCF channel model over using the strongly-typed classes and methods that the WCF service model exposes is that the channel model provides more fine-grained control over the operations that you perform on the Oracle E-Business Suite. This control comes from the fact that in the WCF channel model, you directly control the contents of the messages that you send over the channel.

In certain scenarios, this extra level of control can be beneficial. For example, when you use the WCF channel model to perform an Update operation on a table, you can selectively update columns in the target rows by omitting columns from the update template that you pass in the message. The only columns that are required are those with "nillable=false" in the WSDL. The update method exposed by a WCF client uses a strongly-typed record parameter for the template that includes every column in the table schema.

The topics in this section explain how to perform operations on the Oracle E-Business adapter by using the WCF channel model.

In This Section

- [Overview of Using the WCF Channel Model with the Adapter](#)
- [Creating a Channel](#)
- [Performing an Insert Operation on an Interface Table Using the WCF Channel Model](#)
- [Polling Using SELECT Statement Using the WCF Channel Model](#)

See Also

Other Resources

[Development](#)

Overview of Using the WCF Channel Model with the Adapter

To invoke operations on the Microsoft BizTalk Adapter for Oracle E-Business Suite, your code acts as a WCF client and sends outbound operations to the adapter. In the WCF channel model, your code invokes operations on the adapter by sending a request message over a channel.

To invoke inbound operations, such as receiving polling-based data-changed messages using the **Poll** operation provided by the adapter, your code acts as a WCF service and receives the inbound operation from the adapter. In other words, your code receives a request message from the adapter over a channel.

The topics in this section provide an overview of using the Oracle E-Business adapter with the WCF channel model.

WCF Channel Model Overview

Clients and services communicate by exchanging SOAP messages. The WCF channel model is a low-level abstraction of this message exchange. It provides interfaces and types that enable you to send and receive messages by using a layered protocol stack called a channel stack. Each layer of the stack is composed of a channel, and each channel is created from a WCF binding. At the lowest layer is the transport channel. The transport channel implements the underlying transport mechanism between a service and a client and presents each message to the higher layers (and ultimately the consuming application) as a **System.ServiceModel.Message**. The WCF **Message** class is an abstraction of a SOAP message. WCF provides several channel interfaces, called channel shapes, that model the basic SOAP message exchange patterns, such as request-reply or one-way. A WCF transport binding provides an implementation of one or more channel shapes that higher layers can use to send and receive messages. For more information about the WCF channel model, see "Channel Model Overview" at <http://go.microsoft.com/fwlink/?LinkId=82614>.

The Oracle E-Business adapter is a WCF custom transport binding that exposes an Oracle E-Business Suite artifact as a WCF service.

Supported Channel Shapes for the Oracle E-Business Suite Adapter

The adapter implements the following WCF channel shapes:

- **IRequestChannel (System.ServiceModel.Channels.IRequestChannel)**. The **IRequestChannel** interface implements the client side of a request-reply message exchange. You can use an **IRequestChannel** to perform operations for which you want to consume a response, for example to perform a SELECT query on an interface table.
- **IOutputChannel (System.ServiceModel.Channels.IOutputChannel)**. This shape implements the client side of a one-way message exchange. You can use an **IOutputChannel** to invoke an operation for which you do not need to consume a response, for example to call a procedure that has no OUT parameters.

◆ Important

All underlying calls by the adapter to the Oracle client are synchronous. This includes calls to the Oracle client that are the result of operations invoked over an **IOutputChannel**. When you use an **IOutputChannel**, the adapter discards the response received from the Oracle client.

- **IInputChannel (System.ServiceModel.Channels.IInputChannel)**. This shape implements the service side of a one-way message exchange. You use an **IInputChannel** to receive inbound messages from the adapter.

Like any WCF binding, the Oracle E-Business adapter uses a factory pattern to provide channels to application code. You use a **Microsoft.Adapters.OracleEBSBinding** object to create instances of:

- **System.ServiceModel.ChannelFactory<IRequestChannel>** to provide **IRequestChannel** channels you can use to invoke request-response operations on the adapter.
- **System.ServiceModel.ChannelFactory<IOutputChannel>** to provide **IOutputChannel** channels you can use to invoke one-way operations on the adapter.
- **System.ServiceModel.IChannelListener<IInputChannel>** to provide **IInputChannel** channels you can use to receive inbound messages from the adapter.

Creating Messages for the Oracle Enterprise Business Solution in the WCF Channel Model

In WCF the **System.ServiceModel.Channels.Message** class provides an in memory representation of a SOAP message. You create a **Message** instance by invoking the static **Message.Create** method.

There are two important parts to the SOAP message that you must specify when you create a **Message** instance to send to the Oracle E-Business adapter.

- The message action is a string that is part of the SOAP message header. The message action identifies the operation that should be invoked on the Oracle E-Business Suite. The following shows the message action specified to invoke the **Customer Interface** concurrent program under the **Receivables** application in Oracle E-Business Suite:

`ConcurrentPrograms/AR/RACUST`.

- The message body contains the parameter data for the operation. The message body is composed of well-formed XML that corresponds to the message schema expected by the Oracle E-Business adapter for the requested operation. The following message body specifies a request message to invoke the **Customer Interface** concurrent program.

```
<RACUST xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/ConcurrentPrograms/AR">
  <Description>Customer Interface Program</Description>
  <StartTime></StartTime>
  <CREATE_RECIPROCAL_CUSTOMER>Yes</CREATE_RECIPROCAL_CUSTOMER>
  <ORG_ID>203</ORG_ID>
</RACUST>
```

For information about the Oracle E-Business adapter message schemas and message actions for operations, see [Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#).

The **Create** method is overloaded and offers many different options for providing the message body. The following code shows how to create a **Message** instance by using an **XmlReader** to supply the message body. In this code, the message body is read from a file.

```
XmlReader readerIn = XmlReader.Create("ConcProgRequest.xml");
Message messageIn = Message.CreateMessage(MessageVersion.Default,
    "ConcurrentPrograms/AR/RACUST",
    readerIn);
```

where, `ConProgRequest.xml` contains the request message.

◆ Important

You must provide a message action in your **Message** instance. This is typically done when the **Message** instance is created.

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Creating a Channel

In the WCF channel model, you invoke operations on the Oracle E-Business Suite and receive the results by exchanging SOAP messages with the Microsoft BizTalk Adapter for Oracle E-Business Suite over a WCF channel.

- You invoke operations (outbound operations) by using either an **IRequestChannel** or an **IOutputChannel** to send messages to the adapter.
- You receive messages for inbound operations over an **IInputChannel**.

The topics in this section provide information about how to create and configure channel shapes that are used for inbound and outbound operations.

Creating Outbound (Client) Channels

You can use either an **IRequestChannel** or an **IOutputChannel** to invoke operations on the Oracle E-Business Suite. In either case, you first create a **System.ServiceModel.ChannelFactory** using the appropriate interface. You then use the factory to create the channel. After you have created the channel you can use it to invoke operations on the adapter.

To create and open an outbound channel

1. Create and initialize an instance of **ChannelFactory** for the desired channel shape by using an endpoint and a binding. The endpoint specifies an Oracle E-Business Suite connection URI and the binding is an instance of **OracleEBSBinding**.
2. Provide Oracle E-Business Suite credentials for the channel factory by using the **Credentials** property.
3. Open the channel factory.
4. Get an instance of the channel by invoking the **CreateChannel** method on the channel factory.
5. Open the channel.

You can specify the binding and endpoint address in your code or from configuration.

Specifying the Binding and Endpoint Address in Code

The following code example shows how to create an **IRequestChannel** by specifying the binding and endpoint address in code. The code to create an **IOutputChannel** is the same except that you must specify an **IOutputChannel** interface for the **ChannelFactory** and channel type.

```
// Create binding -- set binding properties before you open the factory.
OracleEBSBinding binding = new OracleEBSBinding();

// Create address
EndpointAddress address = new EndpointAddress("oracleebs://<oracleebs_instance_name>/");

// Create channel factory from binding and address.
ChannelFactory<IRequestChannel> factory =
    new ChannelFactory<IRequestChannel>(binding, address);

// Specify credentials.
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open factory
factory.Open();

// Get channel and open it.
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

Specifying the Binding and Endpoint Address in Configuration

The following code example shows how to create a channel factory from a client endpoint specified in configuration.

```

// Create channel factory from configuration.
ChannelFactory<IRequestChannel> factory =
new ChannelFactory<IRequestChannel>("MyRequestChannel");

// Specify credentials.
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open the factory.
factory.Open();

// Get a channel and open it.
IRequestChannel channel = factory.CreateChannel();
channel.Open();

```

The Configuration Settings

The following code shows the configuration settings used for the preceding example. The contract for the client endpoint must be "System.ServiceModel.Channels.IRequestChannel" or "System.ServiceModel.Channels.IOutputChannel" depending on the kind of channel shape that you want to create.

```

<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <oracleEBSBinding>
        <binding openTimeout="00:05:00" name="OracleEBSBinding" closeTimeout="00:01:00"
Database"
receiveTimeout="00:10:00" sendTimeout="00:01:00" clientCredentialType="
inboundOperationType="Polling" metadataPooling="true" statementCachePur
ge="false"
statementCacheSize="10" pollWhileDataFound="false" pollingInterval="30"
useOracleConnectionPool="true" minPoolSize="1" maxPoolSize="100"
incrPoolSize="5" decrPoolSize="1" connectionLifetime="0" acceptCredenti
alsInUri="false"
useAmbientTransaction="true" notifyOnListenerStart="true"
notificationPort="-1" dataFetchSize="65536" longDatatypeColumnSize="0"
skipNilNodes="true" maxOutputAssociativeArrayElements="32"
enableSafeTyping="false" insertBatchSize="20" useSchemaInNameSpace="tru
e"
enableBizTalkCompatibilityMode="true" enablePerformanceCounters="false"
>
        <mlsSettings language="" dateFormat="" dateLanguage="" numericCharacter
s=""
sort="" territory="" comparison="" currency="" dualCurrency=""
ISOCurrency="" calendar="" lengthSemantics="" nCharConversionExcept
ion="true"
timestampFormat="" timeStampTZFormat="" timeZone="" />
      </binding>
    </oracleEBSBinding>
  </bindings>
  <client>
    <endpoint address="oracleEBS://oracle_ebs_instance/" binding="oracleEBSBinding"
bindingConfiguration="OracleEBSBinding" contract="System.ServiceModel.Chann
els.IRequestChannel"
name="MyRequestChannel" />
  </client>
</system.serviceModel>
</configuration>

```

Creating Inbound (Service) Channels

You configure the Oracle E-Business adapter to poll the Oracle database tables and views by setting binding properties on an instance of **OracleEBSBinding**. You then use this binding to build a channel listener from which you can get an **IInputChannel** channel to receive inbound operations from the adapter.

To create and open an `IInputChannel` to receive messages for inbound operations

1. Create an instance of **OracleEBSBinding**.
2. Set the binding properties required for the inbound operation. For example, for a polling operation, at a minimum you must set the **InboundOperationType**, **PolledDataAvailableStatement**, **PollingAction**, and the **PollingInput** binding properties to configure the Oracle E-Business adapter to poll the Oracle database.
3. Create a binding parameter collection using the **BindingParameterCollection** class and set the credentials.
4. Create a channel listener by invoking **BuildChannelListener<IInputChannel>** method on the **OracleEBSBinding**. You specify the Oracle connection URI as one of the parameters to this method.
5. Open the listener.
6. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on listener.
7. Open the channel.

The following code shows how to create a channel listener and get an **IInputChannel** to receive messages for inbound operations from the adapter.

◆ Important

The Oracle E-Business adapter only supports one-way receive. So, you must use **IInputChannel** to receive messages for inbound operations from Oracle E-Business Suite.

```
// Create a binding: specify the InboundOperationType, the PolledDataAvailableStatement, the
// the PollingInput binding properties.
OracleEBSBinding binding = new OracleEBSBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE";
binding.PollingAction = "InterfaceTables/Poll/FND/APPS/MS_SAMPLE_EMPLOYEE";
binding.PollingInput = "SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE";

// Create a binding parameter collection and set the credentials
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.UserName = "myuser";
credentials.UserName.Password = "mypassword";

BindingParameterCollection bindingParams = new BindingParameterCollection();
bindingParams.Add(credentials);

// Get a listener from the binding and open it.
Uri connectionUri = new Uri("oracleebs://oracle_ebs_instance/");
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>(conn
ectionUri, bindingParams);
listener.Open();

// Get a channel from the listener and open it.
IInputChannel channel = listener.AcceptChannel();
channel.Open();
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Performing an Insert Operation on an Interface Table Using the WCF Channel Model

The Oracle E-Business adapter discovers a set of Insert, Select, Update, and Delete operations on Oracle E-Business Suite interface tables. By using these operations, you can perform simple Insert, Select, Update, and Delete statements qualified by a Where clause on a target interface table. This topic provides instructions on how to perform an Insert operation on an interface table using the WCF channel model.

For more information on how the adapter supports these operations, see [Operations on Interface Tables and Interface Views](#). For more information about how to perform operations on Oracle E-Business Suite using the WCF Channel model, see [Overview of Using the WCF Channel Model with the Adapter](#).

About the Examples Used in this Topic

The example in this topic performs operations on the MS_SAMPLE_EMPLOYEE interface table. The table is created by running the script provided with the samples. For more information about samples, see [Samples](#). A sample, **InsertOperation**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

The Insert Message

To perform operations on the Oracle E-Business Suite using the WCF channel model, you must have the request message specific to the operation. The request message to perform an Insert operation on the MS_SAMPLE_EMPLOYEE interface table resembles the following:

```
<Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/FND/APPS/MS_S
AMPLE_EMPLOYEE">
  <RECORDSET>
    <InsertRecord xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/TableViewRecord/APP
S/MS_SAMPLE_EMPLOYEE">
      <EMP_NO>10050</EMP_NO>
      <NAME>John Smith</NAME>
      <DESIGNATION>Manager</DESIGNATION>
      <SALARY>500000</SALARY>
      <JOIN_DATE>1999-05-31</JOIN_DATE>
    </InsertRecord>
  </RECORDSET>
</Insert>
```

This request message inserts a record with following details:

```
Employee Number = 10050
Name = Tom Smith
Designation = Manager
Salary = 500000
```

You must copy the message to a file, e.g. InsertRequest.xml. This file is used in this example to send the request message to Oracle E-Business Suite using the Oracle E-Business adapter. For more information about the message schema for operations on table, see [Message Schemas for Insert, Update, Delete, and Select Operations](#).

Creating a WCF Channel Application

This section provides instructions on how to create a WCF channel application to perform an Insert operation on the MS_SAMPLE_EMPLOYEE interface table.

To create a WCF channel application for inserting records into the table

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS`, `Microsoft.ServiceModel.Channels`, `System.ServiceModel`, and `System.Runtime.Serialization`.
3. Open the Program.cs file and add the following namespaces:
 - `Microsoft.Adapters.OracleEBS`

- Microsoft.ServiceModel.Channels
- System.ServiceModel
- System.ServiceModel.Channels
- System.Xml

4. Create the binding and endpoint.

```
OracleEBSBinding binding = new OracleEBSBinding();
EndpointAddress address = new EndpointAddress("oracleebs://ebs_instance_name");
```

5. Because you are performing an operation on an interface table, you must set the application context. In this example, to set the application context, you specify the **OracleUserName**, **OraclePassword**, and **OracleEBSResponsibilityName** binding properties. For more information about application context, see [Setting Application Context](#).

```
binding.OracleUserName = "myOracleEBSUserName";
binding.OraclePassword = "myOracleEBSPassword";
binding.OracleEBSResponsibilityName = "myOracleEBSResponsibility";
```

6. Create and open the channel factory. This application sends request message to Oracle E-Business Suite and receives a response, hence you must implement the IRequestChannel interface.

```
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding,
address);
factory.Credentials.UserName.UserName = "<Enter user name here>";
factory.Credentials.UserName.Password = "<Enter password here>";
factory.Open();
```

7. Create and open the channel.

```
IRequestChannel channel;
try
{
    channel = factory.CreateChannel();
    channel.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception :" + ex.Message);
    throw;
}
```

8. Create and send the request message.

```
XmlReader readerIn;
try
{
    readerIn = XmlReader.Create("InsertRequest.xml");
    Console.WriteLine("Reader created");
}
catch (Exception ex)
```

```

{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
Message messageIn;
Message messageOut;
try
{
    messageIn = Message.CreateMessage(MessageVersion.Default, "InterfaceTables/Insert/F
ND/APPS/MS_SAMPLE_EMPLOYEE", readerIn);
    messageOut = channel.Request(messageIn);
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

```

While creating the request message, you must specify the message action that indicates the action that the adapter performs on the interface table. To perform an Insert operation on the MS_SAMPLE_EMPLOYEE table, the message action is `InterfaceTables/Insert/FND/APPS/MS_SAMPLE_EMPLOYEE`. For information about how you can determine the message action for various operations on tables, see [Message Schemas for Insert, Update, Delete, and Select Operations](#).

9. Get the response message.

```

XmlReader readerOut = messageOut.GetReaderAtBodyContents();
XmlDocument doc = new XmlDocument();
doc.Load(readerOut);
doc.Save("C:\\Response.xml");

```

10. Close the message, channel, and channel factory.

```

messageOut.Close();
channel.Close();
factory.Close();

```

11. Build the project. After building the project, you must copy the request message, `InsertRequest.xml`, at the same location as your project executable. Typically, this location is `\\bin\\Debug\\` under your project directory.
12. Run the application. The response message, `Response.xml`, is saved at the location you specified in the application. The response message contains the number of records inserted and resembles the following:

```

<InsertResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/InterfaceTables/
FND/APPS/MS_SAMPLE_EMPLOYEE">
  <InsertResult>1</InsertResult>
</InsertResponse>

```

The value "1" denotes that a single record is inserted into the MS_SAMPLE_EMPLOYEE table.

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Polling Using SELECT Statement Using the WCF Channel Model

You can configure the Oracle E-Business adapter to receive periodic data-change messages by using a SELECT statement to continuously poll the interface tables, interface views, tables and views in Oracle E-Business Suite. You can specify a SELECT statement as a polling statement that the adapter executes periodically to poll Oracle E-Business Suite. You can also specify a post-poll PL/SQL code block that the adapter executes after the polling statement is executed.

To enable polling, you must specify certain binding properties as described in this topic. For more information about how the adapter supports polling, see [Support for Inbound Calls Using Polling](#).

Configuring a Polling Operation with Oracle E-Business Suite Adapter Binding Properties

The following table summarizes the Oracle E-Business adapter binding properties that you use to configure the adapter to receive data change messages. You must specify these binding properties while running the polling application.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. Only if a record is available, the SELECT statement you specify for the PollingInput binding property will be executed.
PollingInterval	Specifies the interval, in seconds, at which the Oracle E-Business adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter sleeps for the remaining time in the interval.
PollingInput	Specifies the polling statement. To poll using a SELECT statement, you must specify a SELECT statement for this binding property. The default is null. You must specify a value for PollingInput binding property to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.
PollingAction	Specifies the action for the polling operation. You can determine the polling action from the service interface generated for the operation using the Add Adapter Service Reference Visual Studio Plug-in.
PostPollStatement	Specifies a statement block that is executed after the statement specified by the PollingInput binding property is executed.
PollWhileDataFound	Specifies whether the Oracle E-Business adapter ignores the polling interval and continuously executes the polling statement, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the polling statement at the specified polling interval. Default is false.

For a more complete description of these properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For a complete description of how to use the Oracle E-Business adapter to poll the Oracle database, read the remainder of this topic.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the Oracle E-Business adapter supports receiving data change messages using SELECT statements, you poll the **MS_SAMPLE_EMPLOYEE** interface table in the **Application Object Library** application. This table is created when you run the create_apps_artifacts.sql script provided with the samples to create these objects in Oracle E-Business Suite.

To demonstrate a polling operation, we do the following:

- Specify a SELECT statement for the **PolledDataAvailableStatement** binding property to determine where the interface table being polled (MS_SAMPLE_EMPLOYEE) has any data. In this example, you can set this binding property as:

```
SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE
```

This ensures that the adapter executes the polling statement only when the MS_SAMPLE_EMPLOYEE interface table has some records.

- Specify a SELECT statement for the **PollingInput** binding property. This statement retrieves all the rows in the MS_SAMPLE_EMPLOYEE interface table. In this example, you can set this binding property as:

```
SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE
```

Note

For information about the FOR UPDATE clause used in the SELECT statement, see [Specifying a FOR UPDATE Clause in the Polling Statement](#).

- Specify a DELETE statement as part of the **PostPollStatement** binding property. This statement will delete all data from MS_SAMPLE_EMPLOYEE interface table. In this example, you can set this binding property as:

```
DELETE FROM MS_SAMPLE_EMPLOYEE
```

After this happens, the next time the statement specified for **PollingInput** will be executed, it will not fetch any data.

- Until more data is added to the MS_SAMPLE_EMPLOYEE interface table, you will not get any polling messages so you must repopulate the MS_SAMPLE_EMPLOYEE interface table with new records. You can do so by running the insert_apps_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table.

Consuming the Polling Request Message

The adapter invokes the polling operation on your code to poll the Oracle E-Business Suite. That is, the adapter sends a polling request message that you receive over an `IInputChannel` channel shape. The polling request message contains the result set of the query specified by the **PollingInput** binding property. You can consume the polling message in one of two ways:

- To consume the message using node-value streaming, you must call the **WriteBodyContents** method on the response message and pass it an **XmlDictionaryWriter** that implements node-value streaming.
- To consume the message using node streaming, you can call **GetReaderAtBodyContents** on the response message to get an **XmlReader**.

About the Examples Used in this Topic

The examples in this topic poll the MS_SAMPLE_EMPLOYEE interface table. A script to generate the table is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **SelectPolling_ChannelModel**, which is based on this topic, is also provided with the Oracle E-Business adapter samples.

Receiving Inbound Messages for Polling Operation Using the WCF Channel Model

This section provides instructions on how to write a .NET application (channel model) to receive inbound polling messages using the Oracle E-Business adapter.

To receive polling messages from the adapter

1. Create a Microsoft Visual C#® project in Visual Studio. For this topic, create a console application.

2. In the Solution Explorer, add reference to `Microsoft.Adapters.OracleEBS`, `Microsoft.ServiceModel.Channels`, `System.ServiceModel`, and `System.Runtime.Serialization`.

3. Open the Program.cs file and add the following namespaces:

- `Microsoft.Adapters.OracleEBS`
- `System.ServiceModel`
- `System.ServiceModel.Description`
- `System.ServiceModel.Channels`
- `System.Xml`

4. Specify a connection URI. For more information about the adapter connection URI, see [The Oracle E-Business Suite Connection URI](#).

```
Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name");
```

5. Create an instance of **OracleEBSBinding** and set the binding properties required to configure polling. At a minimum you must set the **InboundOperationType**, **PolledDataAvailableStatement**, **PollingInput**, and **PollingAction** binding properties. For more information about binding properties used to configure polling, see [Support for Inbound Calls Using Polling](#).

```
OracleEBSBinding binding = new OracleEBSBinding();  
binding.InboundOperationType = InboundOperation.Polling;  
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM MS_SAMPLE_EMPLOYEE";  
binding.PollingInput = "SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE";  
binding.PollingAction = "InterfaceTables/Poll/FND/APPS/MS_SAMPLE_EMPLOYEE";  
binding.PostPollStatement = "DELETE FROM MS_SAMPLE_EMPLOYEE";
```

6. Because you are polling an interface table, you must also set the applications context. For more information about application context and binding properties required for setting application context, see [Setting Application Context](#).

```
binding.OracleUserName = "<Enter user name here>";  
binding.OraclePassword = "<Enter password here>";  
binding.OracleEBSResponsibilityName = "<Enter responsibility here>";
```

7. Create a binding parameter collection and set the credentials.

```
ClientCredentials credentials = new ClientCredentials();  
credentials.UserName.UserName = "<Enter user name here>";  
credentials.UserName.Password = "<Enter password here>";  
  
BindingParameterCollection bindingParams = new BindingParameterCollection();  
bindingParams.Add(credentials);
```

8. Create a channel listener and open it. You create the listener by invoking **BuildChannelListener<IInputChannel>** method on the **OracleEBSBinding**.

```
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>  
(connectionUri, bindingParams);  
listener.Open();
```

9. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on the listener and open it.

```
IInputChannel channel = listener.AcceptChannel();
channel.Open();
```

10. Invoke **Receive** on the channel to get the next inbound message from the adapter.

```
Message message = channel.Receive();
```

11. Consume the result set returned by the inbound operation. You can consume the message using either an **XmlReader** or an **XmlDictionaryWriter**.

```
XmlReader reader = message.GetReaderAtBodyContents();
```

12. Close the channel when you have completed processing the request.

```
channel.Close()
```

◆Important

You must close the channel after you have finished processing the inbound operation. Failure to close the channel may affect the behavior of your code.

13. Close the listener when you are finished receiving data-changed messages.

```
listener.Close()
```

◆Important

Closing the listener does not close channels created using the listener. You must explicitly close each channel created using the listener.

Example

The following example shows a polling application that polls the MS_SAMPLE_EMPLOYEE interface table. The **PollingInput** property contains the select statement that reads all the data from the MS_SAMPLE_EMPLOYEE table and the post poll statement deletes all the data from the same table. The polling message is written to C:\PollingOutput.xml.

Subsequent polling messages will not contain any records until more data is added to the MS_SAMPLE_EMPLOYEE interface table. You can do so by running the insert_apps_data.sql script provided with the samples. After you run this script, the next polling operation will fetch the new records inserted into the table. The adapter will continue to poll until you close the service host by pressing <RETURN>.

```
using System;
using Microsoft.Adapters.OracleEBS;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Xml;

namespace SelectPolling_ChannelModel
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Sample started. This sample will poll 5 times and will perfo
```

```

rm the following tasks:");
    Console.WriteLine("Press any key to start polling...");
    Console.ReadLine();
    IChannelListener<IInputChannel> listener = null;

    IInputChannel channel = null;

    try
    {
        TimeSpan messageTimeout = new TimeSpan(0, 0, 30);

        OracleEBSBinding binding = new OracleEBSBinding();
        binding.InboundOperationType = InboundOperation.Polling;
        binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM MS_SAMPLE_EMP
LOYEE";
        binding.PollingInput = "SELECT * FROM MS_SAMPLE_EMPLOYEE FOR UPDATE";
        binding.PollingAction = "InterfaceTables/Poll/FND/APPS/MS_SAMPLE_EMPLOYEE";
        binding.PostPollStatement = "DELETE FROM MS_SAMPLE_EMPLOYEE";
        binding.OracleUserName = "<Enter user name here>";
        binding.OraclePassword = "<Enter password here>";
        binding.OracleEBSResponsibilityName = "<Enter responsibility here>";

        Uri ConnectionUri = new Uri("oracleebs://ebs_instance_name?");

        ClientCredentials credentials = new ClientCredentials();
        credentials.UserName.UserName = "<Enter user name here>";
        credentials.UserName.Password = "<Enter password here>";

        BindingParameterCollection bindingParams = new BindingParameterCollection()
;
        bindingParams.Add(credentials);

        listener = binding.BuildChannelListener<IInputChannel>(ConnectionUri, bindi
ngParams);
        listener.Open();

        channel = listener.AcceptChannel();
        channel.Open();

        Console.WriteLine("Channel and Listener opened...");
        Console.WriteLine("\nWaiting for polled data...");
        Console.WriteLine("Receive request timeout is {0}", messageTimeout);

        // Poll five times with the specified message timeout
        // If a timeout occurs polling will be aborted
        for (int i = 0; i < 5; i++)
        {
            Console.WriteLine("Polling: " + i);
            Message message = null;
            XmlReader reader = null;
            try
            {
                //Message is received so process the results
                message = channel.Receive(messageTimeout);
            }
            catch (System.TimeoutException toEx)
            {
                Console.WriteLine("\nNo data for request number {0}: {1}", i + 1, t
oEx.Message);
                continue;
            }
            // Get the query results using an XML reader
            try
            {
                reader = message.GetReaderAtBodyContents();
            }

```


Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server

Most line-of-business (LOB) applications have proprietary interfaces that are accessible to relatively few people with the required subject matter expertise. However, because information workers need relevant information at all levels in an organization, the subject matter experts have to export required data into more familiar applications, like Microsoft Excel, for information workers to consume the information.

The new collection of Microsoft Office client applications and features bridges this gap by bringing backend business data into the most common interface for information workers, Microsoft Office. Microsoft Office client applications seamlessly present data from LOB applications through interfaces familiar to information workers. One example is the Business Data Catalog in Microsoft Office SharePoint Server.

In This Section

- [About Business Data Catalog](#)
- [Using the Oracle E-Business Adapter to Integrate the Oracle E-Business Suite with the Business Data Catalog and Microsoft Office SharePoint Server](#)
- [Considerations While Using the Oracle-Business Adapter with Microsoft Office SharePoint Server](#)
- [How Do I Use a Custom Web Part?](#)

See Also

Other Resources

[Development](#)

About Business Data Catalog

The Business Data Catalog is a new integration component in Microsoft Office SharePoint Server 2007 and later. The Business Data Catalog is a shared service and surfaces business data from backend applications via databases and Web services, without the need for writing any code. It is a metadata repository that defines business entities like customers, invoices, and purchase orders. Once defined, clients can use these entities.

The Business Data Catalog provides a Business Data Catalog Definition Editor to help you author application definition files for the Business Data Catalog in Microsoft Office SharePoint Server 2007 and later. Using the Business Data Catalog Definition Editor, you can create and edit metadata for business applications.

For more information about the Business Data Catalog, see "Business Data Catalog" at <http://go.microsoft.com/fwlink/?LinkId=103882>.

See Also

Other Resources

[Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server](#)

Using the Oracle E-Business Adapter to Integrate the Oracle E-Business Suite with the Business Data Catalog and Microsoft Office SharePoint Server

The Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK includes the WCF LOB Adapter Service Development Wizard, which generates a WCF service for specific LOB artifacts. This WCF service is hosted in a hosting environment such as Microsoft Internet Information Services (IIS). The Business Data Catalog Definition Editor uses the URL where the WCF service is hosted to get the Web Services Description Language (WSDL) for the WCF service. Using the WSDL, the Business Data Catalog Definition Editor extracts the methods available to the WCF service. These methods can be used to establish entities and the association between the entities.

The Business Data Catalog Definition Editor helps you create an application definition file (an XML file) that Microsoft Office SharePoint Server can consume. Once the application definition file is imported to Microsoft Office SharePoint Server, you can create Web Parts to present the information from enterprise applications. For more information, see "Creating Web Parts" in Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite in [Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#).

Tutorial

To demonstrate how to use the Oracle E-Business adapter with Microsoft Office SharePoint Server, the BizTalk Adapter Pack includes a tutorial that provides step-by-step instructions to present data from Oracle E-Business Suite on a SharePoint site. See [Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#).

See Also

Other Resources

[Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server](#)

Considerations While Using the Oracle-Business Adapter with Microsoft Office SharePoint Server

This topic contains information about the issues you might encounter while using the Microsoft BizTalk Adapter for Oracle E-Business Suite with Microsoft Office SharePoint Server, along with resolutions. The issues are divided into two categories:

- General issues
- Issues involving custom Web Parts

General Issues

This section contains issues that have no resolution.

Issue 1: The simple type data returned by the WCF service is not displayed

Explanation: Microsoft Office SharePoint Server expects the data returned by the WCF service to be of DataSet or Collection type only. If the data returned by the WCF service is of simple type, Microsoft Office SharePoint Server does not display the data.

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server.

Issue 2: An error message is displayed if the data returned by the WCF service is NULL

Explanation: If the data returned by the WCF service is a NULL value, Microsoft Office SharePoint Server displays an error message. For example, suppose you are using the Business Data List Web Part for the **Finder** method instance, and are searching for customers in Oracle E-Business Suite based on a search expression. The search expression that you specified fetches a NULL value. In this case, Microsoft Office SharePoint Server will display an error message.

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server.

Issue 3: An array of simple type returned by the WCF service is not displayed

Explanation: If the data returned by the WCF service is an array of simple type, Microsoft Office SharePoint Server does not display the data. Moreover, when you execute a method instance in Business Data Catalog Definition Editor that returns an array of simple type, the following error message is displayed: "Backend system adapter returned a structure incompatible with the corresponding metadata (MethodInstance, Parameter or TypeDescriptor)."

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server and Business Data Catalog Definition Editor.

Issue 4: Cannot import an application definition file that contains a complex type parameter having more than 300 fields

Explanation: Microsoft Office SharePoint Server cannot import an application definition file that has more than 300 fields in the complex type parameter returned by the WCF service, and displays an error message if you try to do so. This is due to the limitation of Microsoft Office SharePoint Server of not being able to display more than 300 fields of a complex type parameter.

Resolution: Use the Business Data Catalog Definition Editor to limit the number of fields of the complex type parameter to less than or equal to 300. Depending on your requirement, you can delete the fields of the complex type parameter in the Business Data Catalog Definition Editor that you do not require to be displayed in Microsoft Office SharePoint Server. Alternatively, you can also export the application definition file from Business Data Catalog Definition Editor with all the fields, and then modify the application definition file in a notepad or any XML authoring application to delete the fields that are not required in order to limit the number of fields to 300.

Issues Involving Custom Web Parts

This section contains issues that require the use of custom Web Part for a resolution. For detailed information about using a custom Web Part to resolve issues that might come up while working with Oracle E-Business adapter and Microsoft Office SharePoint Server, see [How Do I Use a Custom Web Part?](#).

Issue 1: Limitation with displaying a single record in Microsoft Office SharePoint Server based on multiple values

Explanation: If you want to display a single record in Microsoft Office SharePoint Server based on multiple values (input parameters) from Oracle E-Business Suite, you cannot use any of the three Web Parts (Business Data List, Business Data Item, and Business Data Related List) specified in Step 3: Create a SharePoint Application to Retrieve Data from Oracle E-Business Suite in [Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#).

Resolution: You must use a custom Web Part to do this. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#). In "Step 1: Create a Custom Web Part" of that topic, you can use the following code sample in step 5. The following code sample takes BankCountry and BankKey as the input parameters, and then displays them as a single record in Microsoft Office SharePoint Server.

```
namespace CustomWebPart
{
    public class CustomWebPart : WebPart
    {
        private string displayText = "Hello World!";

        [WebBrowsable(true), Personalizable(true)]
        public string DisplayText
        {
            get { return displayText; }
            set { displayText = value; }
        }
        protected override void Render(System.Web.UI.HtmlTextWriter writer)
        {
            string BankCountry = "US";
            string BankKey = "134329042";

            /**Step 1: Get the required entity and method.***/

            LobSystem newSystem = ApplicationRegistry.GetLobSystems()["BAPI_BANK_GETDETAIL"
]; // Name specified in application definition file
            LobSystemInstance newSystemInstance = newSystem.GetLobSystemInstances()["BAPI_B
ANK_GETDETAIL_Instance"]; // Name specified in application definition file
            Entity CategoryEntity = newSystem.GetEntities()["Entity"]; // Name specified in
application definition file
            Method newMethod = CategoryEntity.GetMethods()["BAPI_BANK_GETDETAIL"]; // Name
specified in application definition file
            MethodInstance methodInstance = newMethod.GetMethodInstances()["MethodInstance"
]; // Name specified in application definition file

            /**Step 2: Get the list of input parameters.***/

            Object[] args = methodInstance.GetMethod().CreateDefaultParameterInstances(meth
odInstance); //Get the default values of the input parameters.
            Object[] ArgsInput = new Object[args.Length];

            /**Step 3: Assign them required values.***/

            Type t = null;
            char[] myString = BankCountry.ToCharArray();
            String s = new String(myString);
            t = s.GetType();
            ArgsInput[0] = Activator.CreateInstance(t, myString);

            myString = BankKey.ToCharArray();
            s = new String(myString);
            t = s.GetType();
            ArgsInput[1] = Activator.CreateInstance(t, myString);

            /**Step 4: Execute the particular method instance using the required value.***/

            IEntityInstance IE = (IEntityInstance)CategoryEntity.Execute(methodInstance, ne
wSystemInstance, ref ArgsInput); //Method instance of type Specific Finder is being used he
re.

            /**Step 5: Display the output on the custom Web Part in Microsoft Office SharePoint Server
.***/

            writer.Write("<table>");
            writer.Write("<tr>");
            foreach (Field f in CategoryEntity.GetFinderView().Fields)
            {
```

```

        writer.Write("<td>");
        writer.Write(IE[f]);
        writer.Write("</td>");
    }
    writer.Write("</tr>");
    writer.Write("</table>");
}
}

```

Note

The application definition file must contain the **Specific Finder** method instance. A **Specific Finder** method finds a specific record based on an identifier. For information about creating a **Specific Finder** method instance, see "Requirement 2: Retrieve Details for a Specific Customer from the List of Customers" in

[Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts](#) in [Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site](#).

Issue 2: Cannot specify values to array elements

Explanation: If the input parameter of the WCF service is an array, you cannot specify values to the array elements using filters that are defined in the application definition file created using the Business Data Catalog Definition Editor. It implies that you cannot use the Business Data List or Business Data Item Web Part in Microsoft Office SharePoint Server to specify values for these input parameters (elements of array) to the WCF service. This is because of the way arrays are defined in the application definition file.

Resolution: Use a custom Web Part to assign values to array elements. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#). For example, you can use the following code sample in step 3 in "Issue 1: Limitation with displaying a single record in Microsoft Office SharePoint Server based on multiple values" to assign values to array elements.

```

/**Assign required values to parameters of type array.***/
/**Assumption is that the ith parameter of Method is of type Array and all the simple type
elements in the array are of string type.***/

    Type t = asm.GetType(args[i].GetType().ToString()); // Get type of the parameter
    Type TElement = t.GetElementType(); // Getting type of element of array
    int index = 5; //Size of Array
    Array ElementArray = Array.CreateInstance(TElement, index); //Creating an array
of length: index

    for (int ind = 0; ind < index; ind++)
    {
        //Creating an instance of an element of array
        object ElementType = Activator.CreateInstance(TElement);
        FieldInfo[] FI = ElementType.GetType().GetFields();
        for (int f = 0; f < FI.Length; f++)
        {
            ElementType.GetType().GetFields()[f].SetValue(ElementType, (Object)"ElementValue");
        }
        ElementArray.SetValue(ElementType, ind);
    }

    ArgsInput[i] = (object)ElementArray; // As shown in sample, ArgsInput is fed as
input while executing Method Instance

```

Issue 3: Limitation with specifying NULL values to complex type parameters

Explanation: If you do not specify any value for a complex type parameter from a Web Part in Microsoft Office SharePoint Server, NULL should be passed on as the value for the complex type parameter to the WCF service. However, a non-NULL value is passed for the complex type parameter, and NULL is passed for its children elements (of simple type). This causes a mismatch between the expected message schema and the message schema that is passed on to the WCF service. As a result, the Oracle E-Business adapter might display an error message.

Note

To find out the default value of a complex type parameter when no value is passed from a Web Part in Microsoft Office SharePoint Server, use step 2 in the code sample mentioned in "Issue 1: Limitation with displaying a single record in Microsoft Office SharePoint Server based on multiple values."

Resolution: Use a custom Web Part to assign a NULL value to the complex type parameter. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#).

See Also

Other Resources

[Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server](#)

How Do I Use a Custom Web Part?

This section provides information about using a custom Web Part with Microsoft Office SharePoint Server. To use a custom Web Part, you must do the following:

1. Create a custom Web Part
2. Deploy the custom Web Part to a SharePoint portal
3. Configure the SharePoint portal to use the custom Web Part

Before You Begin

Before you create a custom Web Part:

- Publish the Oracle E-Business Suite artifacts as a WCF service. For more information, see [Step 1: Use the Oracle E-Business Adapter to Create and Publish a WCF Service in Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site.](#)
- Create an application definition file for the Oracle E-Business Suite artifacts using the Business Data Catalog in Microsoft Office SharePoint Server. For more information, see [Step 2: Create an Application Definition File for the Oracle E-Business Suite Artifacts in Tutorial: Presenting Data from Oracle E-Business Suite on a SharePoint Site.](#)

Step 1: Create a custom Web Part

To create a custom Web Part using Visual Studio, do the following:

1. Start Visual Studio 2010, and then create a project.
2. In the **New Project** dialog box, from the **Project types** pane, select **Visual C#**. From the **Templates** pane, select **Class Library**.
3. Specify a name and location for the solution. For this topic, specify **CustomWebPart** in the **Name** and **Solution Name** boxes. Specify a location, and then click **OK**.
4. Add a reference to the System.Web component into the project. Right-click the project name in **Solution Explorer**, and then click **Add Reference**. In the **Add Reference** dialog box, select **System.Web** in the **.NET** tab, and then click **OK**. The System.Web component contains the required namespace of System.Web.UI.WebControls.WebParts.
5. Add the required code based on your issue in the project. For the code sample that is relevant to a certain issue, see "Issues Involving Custom Web Parts" in [Considerations While Using the Oracle-Business Adapter with Microsoft Office SharePoint Server.](#)
6. Build the project. On successful build of the project, a .dll file, CustomWebPart.dll, will be generated in the <project folder>/bin/Debug folder.
7. **Only for 64-bit computer:** Sign the CustomWebPart.dll file with a strong name before performing the following steps. Otherwise, you will not be able to import, and hence use the CustomWebPart.dll in the SharePoint portal in "Step 3: Configure the SharePoint Portal to use the custom Web Part." For information about how to sign an assembly with a strong name, see <http://go.microsoft.com/fwlink/?LinkId=197169>.

Step 2: Deploy the custom Web Part to a SharePoint Portal

You must do the following to make the CustomWebPart.dll file (custom Web Part) that is created in "Step 1: Create a custom Web Part" of this topic usable on the SharePoint portal:

- **Copy the CustomWebPart.dll file to the bin folder of the SharePoint Portal:** Microsoft Office SharePoint Server creates portals under the <root drive>:\inetpub\wwwroot\wss\VirtualDirectories folder. A folder is created for each portal, and can be identified with the port number. You must copy the CustomWebPart.dll file created in “Step 1: Create a custom Web Part” of this topic to the <root drive>:\inetpub\wwwroot\wss\VirtualDirectories\<Port_Number>\bin folder. For example, if the port number of your SharePoint portal is 13614, you must copy the CustomWebPart.dll file to the <root drive>:\inetpub\wwwroot\wss\VirtualDirectories\13614\bin folder.

 **Tip**

Another way to find the folder location of your SharePoint portal is by using the **Internet Information Services (IIS) Manager** window (**Start > Run > inetmgr**). Locate your SharePoint portal in the **Internet Information Services (IIS) Manager** window ([computer_name] > Web Sites > [Portal-Name]), right-click, and then click **Properties** in the shortcut menu. In the properties dialog box of the SharePoint portal, click the **Home Directory** tab, and then select the **Local path** box.

- **Add the Safe Control Entry in the web.config File:** Because the CustomWebPart.dll file will be used on different computers and by multiple users, you must declare the file as “safe.” To do so, open the web.config file located in the SharePoint portal folder at <root drive>:\inetpub\wwwroot\wss\VirtualDirectories\<Port_Number>. Under the <SafeControls> section of the web.config file, add the following safe control entry:

- **On 32-bit computer:**

```
<SafeControl Assembly="CustomWebPart" Namespace="CustomWebPart" TypeName="*" Safe="True" />
```

- **On 64-bit computer:**

```
<SafeControl Assembly="CustomWebPart, Version=1.0.0.0, Culture=neutral, PublicKeyToken=<PUBLICKEYTOKEN_OF_CustomWebPart.dll>" Namespace="CustomWebPart" TypeName="*" Safe="True" />
```

Save the web.config file, and then close it.

Step 3: Configure the SharePoint portal to use the custom Web Part

You need to add the custom Web Part to the Microsoft Office SharePoint Server Web Part Gallery, so that you can use it on your SharePoint portal. To do so:

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the Shared Service Provider (SSP) to which you want to add the custom Web Part.
3. On the Shared Services Administration page, in the upper-right corner, click **Site Actions**, and then click **Create**.
4. On the Site Settings page, click **Web Parts** under the **Galleries** column.
5. On the Web Part Gallery page, to add the custom Web Part to the gallery, click **New**. At this point the custom Web Part is not available in the Web Part Gallery page.
6. On the New Web Parts page, locate **CustomWebPart** (name of the custom Web Part) in the list, select the check box on the left, and then click **Populate Gallery** on the top of the page. This will add the **CustomWebPart** entry in the Web Part Gallery page.

Now you can use the custom Web Part (**CustomWebPart**) to create Web Parts in your SharePoint portal. The custom Web

Part (**CustomWebPart**) will appear under the **Miscellaneous** section in the Add Web Parts page.

See Also

Other Resources

[Using the Oracle E-Business Adapter with Microsoft Office SharePoint Server](#)

Samples

Samples for Microsoft BizTalk Adapter for Oracle E-Business Suite are categorized into:

- BizTalk Server samples
- WCF service model samples
- WCF channel model samples
- Microsoft Office SharePoint Server samples

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>. The SQL scripts for creating the interface tables, concurrent programs, tables, and packages used in the samples are also available along with the samples for the Oracle E-Business adapter.

The following tables contain the names and descriptions of the samples for the Oracle E-Business adapter.

BizTalk Server Samples

Sample Directory Name	Description
InterfaceTableInsert	Demonstrates how to insert records into an interface table in Oracle E-Business Suite using Oracle E-Business adapter.
ConcurrentProgram	Demonstrates how to invoke a concurrent program in Oracle E-Business Suite using the Oracle E-Business adapter.
RequestSet	Demonstrates how to invoke a request set in Oracle E-Business Suite using the Oracle E-Business adapter.
MsgContextProperty	Demonstrates how to use the message context properties exposed by the Oracle E-Business adapter to set application context to perform operations on artifacts in Oracle E-Business Suite using the Oracle E-Business adapter.
OracleEBS_CompositeOperation	Demonstrates how to perform composite operations in Oracle E-Business Suite using the Oracle E-Business adapter.
OracleNotifyIncremental	Demonstrates how to receive "incremental" query notification messages from Oracle using the Oracle E-Business adapter.
PollingUsingSelectStatement	Demonstrates how to configure a polling query using a SELECT statement and receive the results using the Oracle E-Business adapter.
PollingUsingStoredProc	Demonstrates how to configure a polling query using a stored procedure and receive the results using the Oracle E-Business adapter.

WCF Service Model Samples

Sample Directory Name	Description
ConcProgram_ServiceModel	Demonstrates how to invoke concurrent programs in Oracle E-Business Suite using the adapter.
ExecuteReader	Demonstrates how to invoke an ExecuteReader operation on Oracle E-Business Suite using the adapter.

Interface_Table_Ops	Demonstrates how to perform operations on interface tables in Oracle E-Business Suite using the adapter.
LargeDataTypes_ServiceModel	Demonstrates how to perform operations on tables with columns of large data types in Oracle E-Business Suite using the adapter.
Notification_ServiceModel	Demonstrates how to receive notifications from databases behind Oracle E-Business Suite using the adapter.
SelectPolling_ServiceModel	Demonstrates how to use a SELECT statement to poll an interface table in Oracle E-Business Suite using the adapter.
StoredProcPolling_ServiceModel	Demonstrates how to use a stored procedure to poll tables in Oracle E-Business Suite using the adapter.

WCF Channel Model Samples

Sample Directory Name	Description
InsertOperation	Demonstrates how to perform an Insert operation on an interface table in Oracle E-Business Suite using the adapter.
SelectPolling_ChannelModel	Demonstrates how to use a SELECT statement to poll an interface table in Oracle E-Business Suite using the adapter.

Microsoft Office SharePoint Server Samples

Sample Directory Name	Description
MOSS_Sample	Demonstrates how to use the Oracle E-Business adapter to create a Windows Communication Foundation (WCF) service from Oracle E-Business Suite artifacts, and then use the WCF service to display data in Microsoft Office SharePoint Server using a Business Data List Web Part.

See Also

Other Resources

[Development](#)

How Do I?

The topics in this section describe how to perform various operations using the Microsoft BizTalk Adapter for Oracle E-Business Suite.

In This Section

- [Configure Transaction Isolation Level and Transaction Timeout](#)
- [Set the Application Context Using Message Context Properties](#)

See Also

Other Resources

[Development](#)

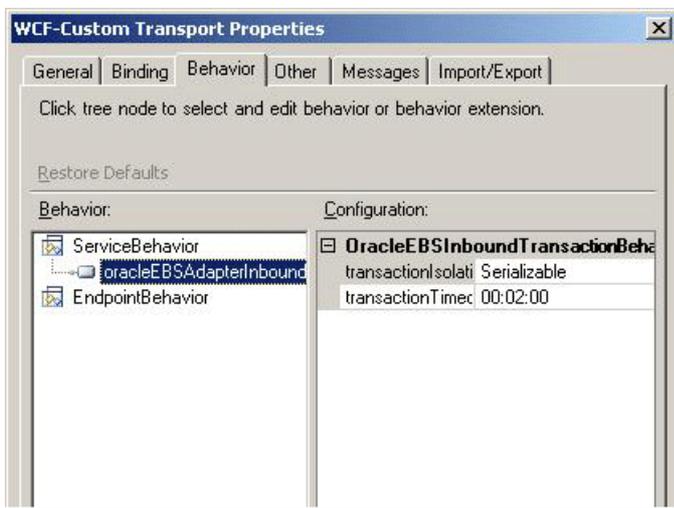
Configure Transaction Isolation Level and Transaction Timeout

While performing inbound operations (Polling and Notification) using the Microsoft BizTalk Adapter for Oracle E-Business Suite with BizTalk Server, you should appropriately configure the transaction isolation level and the transaction timeout values. To do this:

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, and then expand **Applications**.
3. Expand the BizTalk application that you have deployed after generating the metadata using the Oracle E-Business adapter.
4. Right-click **Receive Ports**, point to **New**, and then click **One-way Receive Port**.
5. In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
6. In the left pane of the **Receive Port Properties** dialog box, click **Receive Locations**, and then click **New** in the right pane to define a new receive location.
7. In the **Receive Location Properties** dialog box, click **WCF-Custom** in the **Type** list.
8. Click **Configure** adjacent to the **Type** list.
9. In the **WCF-Custom Transport Properties** dialog box, click the **Behavior** tab.
10. In the **Behavior** list, right-click **ServiceBehavior**, and click **Add extension**.
11. In the **Select Behavior Extension** dialog box, select **oracleEBSAdapterInboundTransactionBehavior**, and click **OK**.
12. In the left pane of the **WCF-Custom Transport Properties**, select the **oracleEBSAdapterInboundTransactionBehavior** service under **ServiceBehavior**.
13. In the right pane of the **WCF-Custom Transport Properties**, specify appropriate values for the **transactionIsolationLevel** and **transactionTimeout** parameters. You can select any of the following transaction isolation levels: **Serializable**, **RepeatableRead**, **ReadCommitted**, **ReadUncommitted**, **Snapshot**, **Chaos**, and **Unspecified**. For information about these transaction isolation levels, see the **Members** section at <http://go.microsoft.com/fwlink/?LinkId=126983>.

◆ Important

Oracle E-Business Suite supports only the following two transaction isolation levels: ReadCommitted and Serializable.



14. Click **OK** in the **WCF-Custom Transport Properties** dialog box.
15. Click **OK** in the open dialog boxes to save the changes.

Set the Application Context Using Message Context Properties

To perform operations on Oracle E-Business Suite artifacts using the Oracle E-Business adapter, you must set the application context appropriately. You can set the application context in the following ways:

- By specifying the binding properties that the adapter exposes. For more information, see [Setting Application Context](#).
- By using message context properties that the adapter exposes. You must consider the following when setting the application context by using message context properties.
 - You can set values only for **ApplicationShortName**, **OrganizationID**, **ResponsibilityKey**, and **ResponsibilityName** by using message context properties. For the user name and password, you must use the binding properties. The value specified for the **ResponsibilityKey** message context property overrides the value specified for the **ResponsibilityName** message context property.
 - If you set the application context using both the binding properties and message context properties, the values specified for message context properties take precedence and override the values specified for the binding properties. However, for example, if you specify the application short name as a message context property, and the organization ID and responsibility name as binding properties, only the value for the application short name is taken from the message context property. The rest are picked from the relevant binding properties.

Why use message context properties over binding properties to set the application context? If you set the application context using binding properties, the WCF-Custom send port for the Oracle E-Business adapter can be used only for the specific organization ID, responsibility, and application that you specified for the binding properties. On the contrary, if you use the message context property, you can configure a "generic" WCF-Custom send port, and set the application context at the message level.

Adapter clients must set the message context properties on the message that is sent to Oracle E-Business Suite to invoke an operation on Oracle E-Business Suite. The messages in BizTalk Server are immutable. Hence, clients must first create a message from the existing message, and then set the message context properties on the new message. For the procedure described in this section, assume that the existing message is called **Request**, and the new message is called **New_Request**.

To set the message context properties for BizTalk applications

1. Open the BizTalk project in Visual Studio.
2. In Solution Explorer, right-click **References**, and then click **Add References**.
3. In the **Add Reference** dialog box, click the **Browse** tab, and then browse to the location where the BizTalk property schema DLL for the Oracle E-Business adapter is available.

This DLL, `Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.dll`, is installed by the BizTalk Adapter Pack at `<installation drive>\Program Files\Microsoft BizTalk Adapter Pack\bin`.
4. Select the DLL, and then click **Add**.
5. In the BizTalk orchestration, add a message, **New_Request**. For the **Message Type** property, make sure you select the same type as the existing request message.
6. Before the Send shape using which the message is sent to the send port, add a Construct Message shape and within that, a Message Assignment shape.
7. Double-click the Message Assignment shape to open **BizTalk Expression Editor**.
8. In **BizTalk Expression Editor**, add the following, and then click **OK**:

```
New_Request = Request;
New_Request(Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ApplicationShortName) =
"AR";
New_Request(Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ResponsibilityKey) = "R
ECEIVABLES_VISION_OPERATIONS";
```

```
New_Request(Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ResponsibilityName) = "
Receivables, Vision Operations (USA)";
New_Request(Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.OrganizationId) = "204"
;
```

◆ Important

The value specified for the **ResponsibilityKey** message context property overrides the value specified for the **ResponsibilityName** message context property.

9. Make sure further processing of the orchestration is done by using the **New_Request** message.
10. Before you can deploy this orchestration in BizTalk Server, you must add the assembly reference for `Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.dll` in the BizTalk application where you will be deploying the orchestration. To deploy an assembly in BizTalk Server:
 - a. Start the BizTalk Server Administration console.
 - b. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then the application to which you want to add a BizTalk assembly.
 - c. Right-click **Resources**, point to **Add**, and then click **BizTalk Assemblies**.
 - d. In the **Add Resources** dialog box, click **Add**, navigate to the folder containing the BizTalk assembly file, which is `<installation drive>\Program Files\Microsoft BizTalk Adapter Pack\bin`. Select the `Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.dll` file, and then click **Open**.
 - e. On the **Options** tab, specify the options for installing the BizTalk assembly to the global assembly cache (GAC), and then click **OK**.

Setting the Language for Performing Operations

The Oracle E-Business adapter supports the Multi-Language Support (MLS) feature of Oracle E-Business Suite, and allows you to specify a language while performing operations. The adapter exposes the **Language** message context property to specify a language for performing operations.

The value specified for the **Language** message context property overrides the value of the **Language** binding property under the **MlsSettings** binding property. For more information about the **MlsSettings** binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

See Also

Other Resources

[How Do I?](#)

Deployment

This section provides information about deploying the Microsoft BizTalk Adapter Pack. It also provides a link to information about troubleshooting common installation issues.

In This Section

- [Installing the BizTalk Adapter Pack](#)
- [Troubleshooting Your Installation](#)

Installing the BizTalk Adapter Pack

See the BizTalk Adapter Pack installation guide for installation prerequisites and complete instructions for installing the BizTalk Adapter Pack. This document is installed on your computer, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

See Also

Concepts

[Troubleshooting Your Installation](#)

Other Resources

[Deployment](#)

Troubleshooting Your Installation

Installation of the Microsoft BizTalk Adapter Pack copies the product binaries on a computer and registers the bindings for each adapter. For information about troubleshooting BizTalk Adapter Pack installation issues, see [Troubleshooting Installation Issues](#).

See Also

Other Resources

[Deployment](#)

Troubleshooting

The BizTalk Adapter Pack uses or depends on several Microsoft technologies, including but not limited to the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, Microsoft Visual Studio, and Microsoft .NET Framework 3.5 SP1 or .NET Framework 4. The BizTalk Adapter Pack is built on top of the WCF LOB Adapter SDK, which in turn requires the .NET Framework 3.5 SP1 or .NET Framework 4. The adapters can be consumed either by writing applications using Visual Studio or by creating BizTalk applications. For issues related to each of these technologies and products, see the respective documentation.

This section provides information about troubleshooting the BizTalk Adapter Pack, including:

- Enabling tracing to diagnose issues with the adapters.
- Handling installation and operational issues that you might encounter when working with the adapters, including probable cause and a resolution.
- Using performance counters to gauge adapter performance.
- Handling exceptions and errors, including probable cause, and a resolution.

In This Section

- [Diagnostic Tracing and Message Logging](#)
- [Troubleshooting Installation Issues](#)
- [Troubleshooting Operational Issues](#)
- [Using Performance Counters](#)
-

Diagnostic Tracing and Message Logging

Diagnostic tracing helps to effectively diagnose problems that you might encounter when using the adapters. This topic provides information about the following three types of tracing supported with Microsoft BizTalk Adapter for Oracle E-Business Suite:

- Oracle server-side tracing using a client identifier.
- WCF tracing between the adapter client and the adapter.
- WCF tracing within the adapter.

Oracle Server Side Tracing Using a Client Identifier

Oracle allows you to perform server-side tracing for the operations performed by client applications on the Oracle database. Because requests from client applications can be routed to different database sessions, it becomes difficult to trace the origin of the request. However, Oracle facilitates end-to-end application tracing using client identifiers. The Oracle E-Business adapter exposes the **OracleConnectionClientId** binding property that allows you to specify the client identifier at the design time for the connection used by the adapter to connect to Oracle. The adapter client identifier helps you in selective tracing of the operations performed by the adapter client on Oracle, and also allows you to filter and view the Oracle server traces based on the client identifier. For information about how you can enable tracing for client identifiers in Oracle, see <http://go.microsoft.com/fwlink/?LinkId=135746>.

WCF Tracing Between the Adapter Client and the Adapter

Adapter clients can enable WCF tracing to trace issues between the adapter client and the adapter. WCF tracing is used to trace the input XML that comes from the adapter client by using the WCF service model and is useful in diagnosing serialization issues. WCF tracing is not used for the WCF channel model or for output messages from the adapter to the adapter client. You can activate WCF tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in `<installation drive>\Program Files\Microsoft Visual Studio <version>\Common7\IDE`.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically `BTSNTSvc.exe.config`. For BizTalk Server 2010, this file is available typically under `<installation drive>\Program Files\Microsoft BizTalk Server 2010`.
 - For a WCF service model .NET application, you must add the excerpt to the app.config file of your project.

To enable WCF tracing, add the following excerpt within the `<configuration>` tag.

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.ServiceModel.MessageLogging"
      switchValue="Verbose, ActivityTracing">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
</system.diagnostics>
```

```

    <source name="System.Runtime.Serialization" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\WCFTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>
<system.serviceModel>
  <diagnostics>
    <messageLogging
      logEntireMessage="true"
      logMalformedMessages="false"
      logMessagesAtServiceLevel="true"
      logMessagesAtTransportLevel="false"/>
  </diagnostics>
</system.serviceModel>

```

This saves the WCF traces to C:\log\WCFTrace.svclog. For more information about WCF tracing, see "Tracing" at <http://go.microsoft.com/fwlink/?LinkId=91242>.

◆ Important

Make sure you mitigate potential security threats of exposing sensitive business data that can be caused when enabling tracing. For recommendations, see [Best Practices](#).

WCF Tracing Within the Adapter

The adapters log different categories of useful information to the trace file such as errors, warnings, and information messages. Such information is useful in understanding the process flow within the adapter and diagnosing issues with the adapter. You can activate the WCF LOB Adapter SDK and adapter tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in <installation drive>\Program Files\Microsoft Visual Studio <version>\Common7\IDE.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically BTSNTSvc.exe.config. For BizTalk Server 2006 R2, this file is available typically under <installation drive>\Program Files\Microsoft BizTalk Server 2006. For BizTalk Server 2010, this file is available typically under <installation drive>\Program Files\Microsoft BizTalk Server 2010.
 - For a WCF service model .NET application, you must add the excerpt to the app.config file of your project.

To enable WCF LOB Adapter SDK and adapter tracing, add the following excerpt within the <configuration> tag.

```

<system.diagnostics>
  <sources>
    <source name="Microsoft.ServiceModel.Channels" switchValue="Error">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="Microsoft.Adapters.OracleEBS" switchValue="Information">
      <listeners>

```

```
        <add name="xml" />
    </listeners>
</source>
</sources>
<sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
traceOutputOptions="LogicalOperationStack"
        initializeData="C:\log\AdapterTrace.svclog" />
</sharedListeners>
<trace autoflush="true" />
</system.diagnostics>
```

This saves the WCF traces to C:\log\AdapterTrace.svclog.

Viewing the Traces

You can use the Windows Communication Foundation (WCF) Service Trace Viewer tool to view the traces. For more information about the tool, see "Using Service Trace Viewer for Viewing Correlated Traces and Troubleshooting" at <http://go.microsoft.com/fwlink/?LinkId=91243>.

Configuring Tracking for BizTalk Applications

The BizTalk Server Administration console lets you configure various tracking options for items such as send ports and receive ports. The tracking configuration settings enable you to track inbound and outbound event data, message properties, message bodies, and orchestrations. For more information about configuring tracking for BizTalk applications, see the "Configuring Tracking Using the BizTalk Server Administration Console" at <http://go.microsoft.com/fwlink/?LinkId=106851>.

You can also use Health and Activity Tracking (HAT) to view historical or tracked data. For more information, see "Viewing Historical or Tracked Data Using HAT" at <http://go.microsoft.com/fwlink/?LinkId=106852>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Installation Issues

This topic was last updated on: June 05, 2009

Installation of the Microsoft BizTalk Adapter Pack copies the product binaries on the computer and registers the bindings for each adapter. This section discusses using troubleshooting techniques to resolve installation errors.

Logging Messages for Setup Actions

The BizTalk Adapter Pack setup program performs the standard task of installing the BizTalk Adapter Pack. Additionally, the setup also performs certain custom actions such as registering the adapter bindings. You can log messages for both the standard as well as custom actions that the setup performs.

- The BizTalk Adapter Pack setup installs the adapter-specific files using an MSI. Therefore, the logging for the setup is the standard MSI logging. For more information about logging for an MSI, see <http://go.microsoft.com/fwlink/?LinkId=196953>.
- All logs for the custom actions that the setup program performs are available at %TEMP%\adaptersetup.log. If the tracing to the log file fails, the traces are also available in the event log.

Known Issues

The following are the most common errors you might encounter when installing the BizTalk Adapter Pack, along with their probable cause and resolution.

- [Setup fails to register adapter bindings](#)
- [Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation](#)
- [Invalid binding error while configuring Oracle E-Business Suite adapter ports in BizTalk Server Administration Console on a 64-bit installation](#)

Setup fails to register adapter bindings

Problem

The Microsoft BizTalk Adapter Pack setup wizard fails to register the adapter bindings, but proceeds with the adapter installation.

Cause

This might result due to problems with Windows Communication Foundation (WCF) installation, Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK installation, or the machine.config file being corrupt. The adapter bindings are written to the machine.config file.

Resolution

You should manually register the Oracle E-Business adapter binding.

To register the adapter binding

1. Navigate to the machine.config file on the computer. For example, on a 32-bit platform, the machine.config is available under <system drive>:\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG.

In this path, <version> is the version of the .NET Framework. For example, for Microsoft .NET Framework 3.5 SP1, the version will be v2.0.50727.

2. Open the file by using a text editor.

3. To register the Oracle E-Business adapter binding:

- a. Search for the element "system.serviceModel" and add the following under it:

```
<client>
  <endpoint binding="oracleEBSBinding" contract="IMetadataExchange" name="oracleebs" />
</client>
```

- b. Search for the element "bindingElementExtensions" under system.serviceModel\extensions.

- c. Look for the missing Oracle E-Business adapter binding. Add the following section under the "bindingElementExtensions" node.

For Oracle E-Business adapter, add:

```
<add name="oracleEBSAdapter" type="Microsoft.Adapters.OracleEBS.OracleEBSBindingElementExtension
```

```
Element, Microsoft.Adapters.OracleEBS, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

- d. Search for the element "bindingExtensions" under system.serviceModel\extensions.
- e. Look for the missing Oracle E-Business adapter binding. Add the following section under the "bindingExtensions" node.

For Oracle E-Business adapter, add:

```
<add name="oracleEBSBinding" type="Microsoft.Adapters.OracleEBS.OracleEBSBindingCollectionElement, Microsoft.Adapters.OracleEBS, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

Note

For information about how to determine the public key and the version, see [Determining the Public Key and Version](#).

4. Save and close the machine.config file.

Determining the Public Key and Version

Perform the following steps to determine the public key for Oracle E-Business adapter.

To determine the public key

1. Navigate to the Windows directory, typically C:\WINDOWS\assembly.
2. Right-click the DLL for which you want the public key and the version, and then select **Properties**. The following table lists the name of the DLL for Oracle E-Business adapter.

Adapter	Name of the DLL
Oracle E-Business adapter	Microsoft.Adapters.OracleEBS

3. On the **General** tab, the value against the **Public Key Token** label specifies the public key for the DLL. Similarly, value against the **Version** label specifies the version number for the DLL.
4. Copy the public key, and then click **Cancel**.

Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation

Problem

Using the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in from Visual Studio on a 64-bit computer running 64-bit version of the BizTalk Adapter Pack results in the following error:

```
No valid adapters are installed on this machine
```

Cause

The Oracle E-Business adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, Visual Studio runs as a 32-bit process and hence when you launch the Consume Adapter Service Add-in from Visual Studio, the plug-in checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

- Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

- Install both the 32-bit and 64-bit versions of the Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see "Oracle Data Provider for .NET FAQ" at <http://go.microsoft.com/fwlink/?LinkId=92834>.

Invalid binding error while configuring Oracle E-Business Suite adapter ports in BizTalk Server Administration Console on a 64-bit installation

Problem

When you try to configure a port for the adapter in BizTalk Server Administration console, you get the following error:

```
"Unable to create binding configuration element for editing. Check the values of the BindingType and BindingConfiguration properties.  
(Microsoft.Biztalk.Adapter.Wcf.Converters.CreateBindingException) Unable to get binding type for binding extension "oracleEBSBinding".  
Verify the binding extension is registered in machine.config."
```

Cause

The Oracle E-Business adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, BizTalk Server Administration console runs as a 32-bit process and hence when you configure a port for the adapter, it checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

- Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

- Install both the 32-bit and 64-bit versions of the Oracle Data Access Components for Oracle Client 11.1.0.6 with Patch Set 11.1.0.7.

Note

To make sure your application works with the most recent version of ODP.NET, you must have the "policy DLLs" installed on the computer and registered in the GAC. For more information, see "Oracle Data Provider for .NET FAQ" at <http://go.microsoft.com/fwlink/?LinkId=92834>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Operational Issues

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve operational errors that you might encounter when using Microsoft BizTalk Adapter for Oracle E-Business Suite.

Enabling Tracing

For more information about tracing support in the Oracle E-Business adapter, [Diagnostic Tracing and Message Logging](#).

Known Issues

The following are the most common errors you might encounter when using the BizTalk Adapter Pack, along with their probable cause and resolution.

- [Error in loading the adapter bindings](#)
- [The Oracle E-Business Suite adapter does not display in the list of adapters in BizTalk Server Administration console](#)
- [Error while performing operations on the Oracle E-Business Suite](#)
- [BizTalk process might crash due to an incorrect Oracle client version when a request message is dropped at the receive location](#)
- [Adapter might throw an overflow exception on executing an operation](#)
- [Adapter might throw an arithmetic overflow exception on executing an ExecuteScalar operation](#)
- [Adapter client might throw the following exception on executing an operation: "Could not retrieve user id, responsibility id, application id. Check if correct values were passed in."](#)
- [Error with RootNode TypeName in BizTalk Projects](#)
- [Invalid binding warning when using the adapter in Visual Studio 2010](#)
- [BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host](#)
- [Timeout Exception while browsing Oracle E-Business Suite Artifacts in Visual Studio](#)
- [Memory usage and thread count increases when using the adapter in a transacted inbound operation](#)

Error in loading the adapter bindings

Problem

When you try to start the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in, you get the following error:

```
There was an error loading the binding, <binding name>, from your system configuration.ConfigurationErrorsException: Exception has been thrown by the target of an invocation.
```

Cause

When you try to start the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, WCF loads the adapter bindings for all the installed adapters. In turn, the adapter bindings are dependent on the specific client software for the enterprise application. You might face this issue for one or both of the following reasons:

- The required LOB client software is not installed on the computer where you installed the adapter.
- You did a Typical or Complete installation of the adapter, which installs all the adapters contained in the BizTalk Adapter Pack. However, the LOB client libraries might be installed for only one enterprise application. As a result, the GUI fails to load the bindings for the other adapters.

Resolution

- Make sure that the required LOB client versions are installed on the computer where you installed the BizTalk Adapter Pack. For information about the supported client versions, see the installation guide available at <installation drive>\Program Files\Microsoft BizTalk Adapter Pack.
- Make sure you do a custom installation of the adapters to install only the adapter you need.

The Oracle E-Business Suite adapter does not display in the list of adapters in BizTalk Server Administration console

Problem

Unlike the earlier version of the adapters shipped with BizTalk Server, the Oracle E-Business adapter shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console.

Cause

The latest Oracle E-Business adapter is a WCF custom binding. So, although the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF custom bindings and hence, does not display the WCF-based Oracle E-Business adapter.

Resolution

You can explicitly add the Oracle E-Business adapter to the BizTalk Server Administration console by following the steps mentioned in [Adding the Oracle E-Business Suite Adapter to BizTalk Server Administration Console](#).

Error while performing operations on the Oracle E-Business Suite

Problem

The adapter gives the following error when performing any operation on the Oracle E-Business Suite using BizTalk Server.

- **For BizTalk Server 2010**

```
System.ArgumentNullException: Value cannot be null.
```

Cause

The WCF action for the message is not specified. WCF requires a SOAP action to be specified for every operation, which informs the adapter about the operation to be performed on the LOB application.

Resolution

Specify the SOAP action in the send port or as a message context property in a BizTalk orchestration. For instructions, see [Specifying SOAP Action](#). See [Technical Reference](#) to see a list of actions for each operation.

BizTalk process might crash due to an incorrect Oracle client version when a request message is dropped at the receive location

Problem

After a request message is dropped at a receive location defined in a BizTalk orchestration, the orchestration consumes the message and BizTalk host (BTSNTSvc.exe) crashes and restarts.

Cause

Installing the Oracle client adds the reference to the latest client assemblies in the PATH variable. Also, the references to the most recent installation of the Oracle client assembly precede the reference to the existing client assemblies. So, if the most recent Oracle client installation is not of a supported client version, BizTalk host crashes and then restarts.

For example, assume that the supported Oracle client 11.1.0.7 is already installed on the computer and the PATH variable has the following reference:

```
C:\oracle\product\11.1.0\client_1\bin;
```

If an unsupported Oracle client, for example 10.2.0.3, is installed on the same computer, the PATH variable will have the following reference:

```
C:\oracle\product\10.2.0\db_2\bin;C:\oracle\product\11.1.0\client_1\bin;
```

Note that the unsupported client version is referenced before the supported version and hence BizTalk host crashes. If there are more than one BizTalk hosts running, then the one hosting the adapter crashes.

Resolution

If more than one Oracle client is installed on the same computer, make sure the supported Oracle client version is referenced before the other Oracle client versions in the PATH variable. For example, if the supported Oracle client version is 11.1.0.7, the reference in the PATH variable must look like:

```
C:\oracle\product\11.1.0\client_1\bin;C:\oracle\product\10.2.0\db_2\bin;
```

Adapter might throw an overflow exception on executing an operation

Problem

Using the adapter, if you try to perform an operation containing Oracle numeric data types inside DataSets or weakly-typed REF CURSORS, the adapter might throw an overflow exception.

Cause

This happens if you supply a large value for the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS that cannot fit into the respective .NET type.

Resolution

If you want to pass large values for the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS, you must enable safe typing by setting the value of the **EnableSafeTyping** binding property to **true**. Enabling safe typing exposes the Oracle numeric data type inside DataSets or weakly-typed REF CURSORS as strings.

Adapter might throw an arithmetic overflow exception on executing an ExecuteScalar operation

Problem

Using the adapter, if you try to execute a SELECT statement in an ExecuteScalar operation that retrieves a large number, the adapter throws the following exception: "System.OverflowException: Arithmetic operation resulted in an overflow."

Cause

This happens due to the known limitation of ExecuteScalar operation in ODP.NET. ODP.NET tries to fit in the data into the .NET Decimal data type, and if the result is too large to fit in the .NET Decimal type, the exception is thrown.

Resolution

Use TO_CHAR() in the SELECT statement in the ExecuteScalar operation to convert the returned data as string.

Adapter client might throw the following exception on executing an operation: "Could not retrieve user id, responsibility id, application id. Check if correct values were passed in."

Problem

The adapter clients might throw this exception if you are performing operations on Oracle E-Business Suite artifacts (interface tables, interface views, concurrent programs, and request sets).

Cause

This happens if you supply an incorrect combination of Oracle user name, password, and responsibility name while performing operations on interface tables, interface views, concurrent programs, and request sets. The Oracle E-Business adapter requires these values in order to set the application context for these artifacts. For more information about setting application context, see [Setting Application Context](#).

Resolution

You must specify a correct combination of the Oracle user name, password, and responsibility to appropriately set application context for an Oracle E-Business Suite artifact. To specify values for Oracle user name and password, you must use the **OracleUserName** and **OraclePassword** binding properties. To specify value for the Oracle responsibility, you can either use the **OracleEBSResponsibilityName** binding property or message context property.

Error with RootNode TypeName in BizTalk Projects

Problem

In a BizTalk project in Visual Studio, if the schemas generated from the Consume Adapter Service Add-in contains invalid characters or reserved words for the **RootNode TypeName** property, the following error will occur while compiling the project:

```
Node <node reference> - Specify a valid .NET type name for this root node.  
The current .NET type name of this root node is invalid (it is a reserved BizTalk Keyword or is an invalid C# identifier).
```

Resolution

1. Right-click the root node referenced in the error and select **Properties**.
2. For the **RootNode TypeName** property, remove any illegal characters or reserved words, for example, dot (.).

Invalid binding warning when using the adapter in Visual Studio 2010

Problem

When you use the adapter to create an application in Visual Studio 2010 and you open the configuration file (app.config) generated by the adapter, you see a warning similar to the following:

```
The element 'bindings' has invalid child element 'oracleEBSBinding'. List of possible elements expected: 'basicHttpBinding, customBinding, ...'
```

Cause

This warning appears because the Oracle E-Business adapter binding, `oracleEBSBinding`, is not a standard binding shipped with the Windows Communication Foundation (WCF).

Resolution

You can safely ignore this warning.

BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host

Problem

BizTalk Server throws an XLANG exception or an exception stating that the application cannot locate the document specification because multiple schemas matched the message type.

Cause

This happens because of either of the following:

- You have generated more than one Notification schema in a BizTalk Server project, deployed it to a BizTalk Server application, and then ran the application to receive notifications from the Oracle database. Because the Notification schemas are common, there is a conflict between the schemas that are deployed in the BizTalk Server application.
- In case of multiple projects, you have generated a Notification schema for each of the BizTalk Server projects, deployed each project to a separate BizTalk Server application on the same host, and then ran an application or applications to receive notifications from the Oracle database. Because the schemas and assemblies are accessible across the applications in BizTalk Server, there is a conflict between the common schemas deployed under various BizTalk Server applications and assemblies.

Resolution

Use a single Notification schema file for a BizTalk Server application. If you need to use the Notification schema in multiple BizTalk Server applications on the same host, create an application containing a single Notification schema, and then use the notification schema from all other applications in BizTalk Server.

Timeout Exception while browsing Oracle E-Business Suite Artifacts in Visual Studio

Problem

While browsing Oracle E-Business Suite artifacts in a Visual Studio project using the Add Adapter Metadata Wizard, Add Adapter Service Reference Plug-in, or Consume Adapter Service Add-in you might encounter a timeout exception.

Cause

This might happen if the server hosting the Oracle E-Business Suite is slow, server is located at a remote location, or the schema you are looking under has a large number of artifacts.

Resolution

You can either choose to increase the value of the **SendTimeout** binding property or provide a search expression in the **Search in category** text box to reduce the number of artifacts that the adapter retrieves.

For more information about specifying binding properties, see [Specifying Binding Properties](#). For more information about searching artifacts in Oracle E-Business Suite, see [Browsing, Searching, and Retrieving Metadata for Oracle E-Business Operations](#).

Memory usage and thread count increases when using the adapter in a transacted inbound operation

Problem

In a transacted inbound operation, such as Polling, **if there is no data available in the table being polled** and the adapter continues to poll, over a period of time you experience an increase in the memory usage and the thread count.

Cause

If there is no data available in the table being polled, after every receive timeout cycle, Windows Communication Foundation (WCF) spawns a new thread to continue the polling operation. Hence, the thread count and memory usage increases over a period of time. However, if the table being polled has some data, the same thread continues to perform all subsequent polls.

Resolution

We recommend setting the **ReceiveTimeout** to the maximum possible value, which is 24.20:31:23.6470000 (24 days) so that a new thread is spawned only every 24 days. This will ensure that the memory usage and thread count does not grow too much too soon.

For more information about the **ReceiveTimeout** binding property, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). For instructions on specifying binding properties, see [Specifying Binding Properties](#).

Note

When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.

See Also

Other Resources

[Troubleshooting](#)

Using Performance Counters

Microsoft BizTalk Adapter Pack clients can use performance counters to gauge the performance of the adapters. The BizTalk Adapter Pack setup program creates the performance counter category **BizTalk .NET Adapter for Oracle E-Business Suite** along with installing the BizTalk Adapter Pack.

The LOB Time (Cumulative) Performance Counter

The **BizTalk .NET Adapter for Oracle E-Business Suite** category has one performance counter called the "LOB Time (Cumulative)." This performance counter denotes the time, in milliseconds, that the LOB client library takes to complete an action that the adapter initiates. The Oracle E-Business adapter creates an instance of the performance counter in any of the following patterns:

```
<process id>:<app domain id>:<oracle data source>:<string>
```

Where "string" could be:

- Connection.Open
- Connection.Close
- Metadata
- Message action. For example, if the action is `InterfaceTables/Insert/FND/APPS/MS_SAMPLE_EMPLOYEE` then the string will be `InterfaceTables.Insert.FND.APPS.MS_SAMPLE_EMPLOYEE`.

The Oracle data source is the same as specified in the connection URI.

The performance counter is initialized only after the adapter makes the first call to the Oracle database. Also, the `InstanceLifetime` property of the performance counter is set to 'Process', which means that the performance counter ceases to exist as soon as the program that creates the counter terminates. For more information about the `InstanceLifetime` property, see <http://go.microsoft.com/fwlink/?LinkId=104181>.

Note

The precision of the LOB Time (Cumulative) performance counter is 16 milliseconds.

Enabling Performance Counters

The performance counters can be enabled or disabled by setting the binding property **EnablePerformanceCounters**. To enable performance counters, set the **EnablePerformanceCounters** binding property to **True**. To disable performance counters, set **EnablePerformanceCounters** to **False**. By default, **EnablePerformanceCounters** is set to **False**.

Performance Counters and the WCF LOB Adapter SDK

Changing the value of the **EnablePerformanceCounters** binding property also changes the value of the corresponding performance counter for the WCF LOB Adapter SDK. Also, the binding property for the WCF LOB Adapter SDK is static, whereas that for the BizTalk Adapter Pack is dynamic. Therefore, if there are two instances of the Oracle E-Business adapter binding in the AppDomain, and the **EnablePerformanceCounters** binding property is set to **True** in one and **False** in the other, the adapter-specific performance counter will be enabled in one and disabled in the other. However, because the binding property for WCF LOB Adapter SDK is static, it will either be set to **True** or **False** depending on what value was specified last.

See Also

Other Resources

[Troubleshooting](#)

Technical Reference

This section covers technical reference information about Microsoft BizTalk Adapter for Oracle E-Business Suite.

In This Section

- [Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)
- [Glossary](#)

Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite

The Microsoft BizTalk Adapter for Oracle E-Business Suite is a Windows Communication Foundation (WCF) custom binding. It exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel.

As a WCF service, the Oracle E-Business adapter exposes metadata for its operations and data types by using standard WCF mechanisms. The sections in this topic describe the XML structure of the messages and data types that the Oracle E-Business adapter uses.

In This Section

- [Basic Oracle Data Types](#)
- [Message Schemas for Insert, Update, Delete, and Select Operations](#)
- [Message Schemas for Stored Procedures, Functions, and PL/SQL APIs](#)
- [Message Schemas for Concurrent Programs](#)
- [Message Schemas for Request Sets](#)
- [Message Schemas for Special LOB Operations](#)
- [Message Schemas for the Polling Operations](#)
- [Message Schemas for the Notification Operation](#)
- [Message Schemas for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#)
- [Message Schemas for the Composite Operation](#)

See Also

Other Resources

[Technical Reference](#)

Basic Oracle Data Types

This topic describes how the Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces basic Oracle data types.

Supported Oracle Data Types

The Oracle E-Business adapter supports safe typing for some Oracle data types. When safe typing is enabled, these data types are represented as strings. You configure safe typing by enabling the **EnableSafeTyping** binding property (disabled by default). For more information about the Oracle E-Business adapter binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#).

Note

Safe typing is not supported if data types are inside User Defined Types (UDTs).

The following table shows how the Oracle data types are surfaced with safe typing disabled (**EnableSafeTyping** is **false**). Oracle data types that are affected by the **EnableSafeTyping** binding property are marked with an asterisk (*).

Oracle Data Type	XSD type	.NET type	Comments
BFile	input: xsd:string output: xsd:base64Binary	String Byte[]	BFile data type is not supported inside complex types (such as RecordType, TableType, UDT, and VArray).
Blob	xsd:base64Binary	Byte[]	-
Char	xsd:string	String	-
Clob	xsd:string	String	-
Date* (No safe typing if inside an UDT)	xsd:dateTime	DateTime	<p>Date values cannot contain time zone information (UTC or UTC offsets):</p> <ul style="list-style-type: none"> • xsd:dateTime values must not contain UTC or UTC offsets • DateTime.Kind must be DateTimeKind.Unspecified <p>If time zone information is specified, the adapter throws an XmlReaderParsingException exception with a message that indicates the field.</p> <p>Note</p> <p>The Oracle E-Business adapter exposes Oracle Date data type as xsd:dateTime instead of xsd:date because:</p> <ul style="list-style-type: none"> • Oracle Date data type can also contain time value. • There is no .NET equivalent for xsd:date.

Float**	xsd:float if prec <=7 xsd:double if prec >7 and <=15 xsd:string if prec >15	Float Double String	You must specify the value consistent with the format specified for the decimal character and group separator in the NumericCharacters binding property under the MlsSettings binding property. If no value is specified for the NumericCharacters binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.
IntervalDS	xsd:string xsd:duration if inside an UDT	String TimeSpan if inside an UDT	The adapter returns the IntervalDS data as a string using the OracleIntervalDS.ToString method. The value should be expressed in Oracle native format: Day HH:MI:SSxFF (for example, "5 15:30:12.99").
IntervalYM	xsd:string xsd:long if inside an UDT	String Long if inside an UDT	The adapter returns the IntervalYM data as a string using the OracleIntervalYM.ToString method. The value should be expressed in Oracle native format: Year-Month; for example, "1-2" (1 year and 2 months).
Long	xsd:string	String	Starting with the Oracle database 9i release, the LONG data type is deprecated. Oracle recommends using the Large Object (LOB) data types instead. Therefore, when performing operations on the Oracle database using the Oracle Database adapter, we recommend using Oracle database artifacts that operate on LOB data types and not the LONG data type.
LongRaw	xsd:base64Binary	Byte[]	-
NChar	xsd:string	String	-
NClob	xsd:string	String	-
Number**	xsd:decimal if prec <=28 xsd:string if prec >28	Decimal String	-
NVarchar2	xsd:string	String	-

Raw	xsd:base64Binary	Byte[]	
RowID	xsd:string	String	-
TimeStamp* (No safe typing if inside an UDT)	xsd:dateTime if prec <= 7 xsd:string if prec > 7	DateTime String	<p>When exposed as string (prec > 7), the value should be expressed in Oracle NLS_TIMESTAMP_FORMAT. You can specify the string format for TimeStamp data types in the TimeStampFormat binding property under the MlsSettings binding property. If no value is specified for the TimeStampFormat binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.</p> <p>TimeStamp values cannot contain time zone information (UTC or UTC offsets):</p> <ul style="list-style-type: none"> • xsd:dateTime values must not contain UTC or UTC offsets • DateTime.Kind must be DateTimeKind.Unspecified <p>If time zone information is specified, the adapter throws an XmlReaderParsingException exception with a message that indicates the field.</p>
TimeStampLTZ	xsd:string	String	<p>TimeStampLTZ is not supported inside UDTs.</p> <p>Outside an UDT: The value should be expressed in Oracle NLS_TIMESTAMP_TZ_FORMAT. You can specify the string format for TimeStampLTZ data types in the TimeStampTZFormat binding property under the MlsSettings binding property. If no value is specified for the TimeStampTZFormat binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.</p>
TimeStampTZ	xsd:string xsd:dateTime if inside an UDT	String DateTime if inside an UDT	<p>Outside an UDT: The value should be expressed in Oracle NLS_TIMESTAMP_TZ_FORMAT. You can specify the string format for TimeStampTZ data types in the TimeStampTZFormat binding property under the MlsSettings binding property. If no value is specified for the TimeStampTZFormat binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.</p>
Decimal**	xsd:decimal if prec <= 28 xsd:string if prec > 28	Decimal String	-
Varchar2	xsd:string	String	-
BinaryFloat**	xsd:float if prec <= 7 xsd:string if prec > 7	Float String	<p>You must specify the value consistent with the format specified for the decimal character and group separator in the NumericCharacters binding property under the MlsSettings binding property. If no value is specified for the NumericCharacters binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.</p>

Binary Double**	xsd:double if prec <= 15 xsd:string if prec > 15	Double String	-
Binary Integer*	xsd:integer	Int32	
Boolean	xsd:boolean	Nullable boolean	
XMLTYPE	xsd:string	String	Supported for top level procedure parameters. Reserved XML characters like '<', '>' must be replaced with their entity representation (&lt; , &gt;) when developing applications in BizTalk, and when using WCF channel Model. This is not required in the case of WCF Service Model.

*The way in which these Oracle data types are surfaced is affected by the **EnableSafeTyping** binding property.

The way in which these Oracle numeric data types inside DataSets and weakly-typed REF CURSORS are surfaced is affected by the **EnableSafeTyping binding property.

◆ Important
<ul style="list-style-type: none"> The maximum length of the value in an Oracle data type in the Oracle E-Business adapter is bound by the maximum length of the value supported by ODP.NET for the Oracle data type. The Oracle E-Business adapter internally treats the Oracle numeric data types inside UDTs as .NET Decimal. However, in general (that is outside UDTs), the Oracle E-Business adapter internally treats the Oracle numeric data types as OracleDecimal.

Safe Typing Enabled

The following table shows how the Oracle data types that are affected by safe typing are changed when the **EnableSafeTyping** binding property is **true**.

📌 Note
Oracle data types that are not in this table are surfaced in the same way whether safe typing is enabled or disabled.

Oracle Data Type	XML Schema Type	.NET Type	Comment

Date	xs	St	The value should be expressed in Oracle NLS_DATE_FORMAT. You can specify the format for the Date data types in the DateFormat binding property under the MlsSettings binding property. If no value is specified for the DateFormat binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.
TimeStamp	xs	St	The value should be expressed in Oracle NLS_TIMESTAMP_FORMAT. You can specify the string format for TimeStamp data types in the TimeStampFormat binding property under the MlsSettings binding property. If no value is specified for the TimeStampFormat binding property, the adapter uses the MLS settings for the ODP.NET client on the same computer where the adapter is installed.

◆ Important

If safe typing is enabled, the Oracle numeric data types inside DataSets and weakly-typed REF CURSORS are always exposed as strings.

Validation

The Oracle E-Business adapter performs no explicit validation on the values that you specify for Oracle data types. However, depending on the Oracle data type and whether safe typing is enabled or disabled, implicit validation may be performed:

- When de-serializing between the XML passed in a message and the .NET types that are used internally by the adapter.
- By ODP.NET for some data types.

See Also

Other Resources

[Technical Reference](#)

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for Insert, Update, Delete, and Select Operations

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces basic Insert, Update, Delete, and Select operations for each interface table in Oracle E-Business Suite and each table in the underlying database. The adapter also surfaces the Select operation for each interface view in Oracle E-Business Suite and each view in the underlying database. These operations perform the appropriate SQL statement qualified by a WHERE clause. The Oracle E-Business adapter uses strongly-typed records and record sets in these operations.

Message Structure for Basic Operations

The following table shows the XML message structure for the basic operations exposed by the Oracle E-Business adapter on Oracle E-Business Suite interface tables and interface views and on the underlying database tables and views. The target object for an operation is specified in the message action and also appears in the target namespace.

 Note
See attribute descriptions after table.

Operation	XML Message	Description	SQL Executed by the Adapter
Insert	<pre><Insert xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <RECORDSET> <InsertRecord> <[FIELD1_NAME] InlineValue="value">[value1]</[FIELD1_NAME]> <[FIELD2_NAME] InlineValue="value">[value2]</[FIELD2_NAME]> ... </InsertRecord> </RECORDSET> </Insert></pre>	The value for the InlineValue attribute, if specified, overrides the value of an element.	<pre>INSERT INTO TABLE_NAME (FIELD1_NAME, FIELD2_NAME, ...) VALUES (value1, value2, ...);</pre>
Insert Response	<pre><InsertResponse xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <InsertResult>[rows inserted]</InsertResult> </InsertResponse></pre>	The number of rows inserted is returned in the InsertResult element.	--

Select	<pre><Select xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <COLUMN_NAMES>[COLUMN_list]</COLUMN_NAMES> <FILTER>WHERE_clause</FILTER> </Select></pre>	<p>A SELECT query is performed on the target table using the WHERE clause specified in the FILTER element. The result set contains the columns in the comma-separated list of column names specified in the COLUMN_NAMES element.</p> <p>◆ Important This is the only operation that is applicable for interface views and database views.</p>	<pre>SELECT COLUMN_list FROM TABLE_NAME WHERE WHERE_clause;</pre>
Select Response	<pre><SelectResponse xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <SelectResult> <SelectRecord> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </SelectRecord> </SelectResult> </SelectResponse></pre>	<p>The result set generated by the SELECT query.</p>	<pre>--</pre>
Update	<pre><Update xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <RECORDSET> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </RECORDSET> <FILTER>WHERE_clause</FILTER> </Update></pre>	<p>Rows that match the where clause specified in the FILTER element are updated to the values specified in the RECORDSET. Only the columns that are specified in the RECORDSET element are updated in each matching row.</p>	<pre>UPDATE [TABLE_NAME] SET [FIELD1_NAME] = value1, [FIELD2_NAME] = value2, ... WHERE WHERE_clause;</pre>
Update Response	<pre><UpdateResponse xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <UpdateResult>[rows inserted]</UpdateResult> </UpdateResponse></pre>	<p>The number of rows updated is returned in the UpdateResult element.</p>	<pre>--</pre>

Delete	<pre><Delete xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <FILTER>WHERE_clause</FILTER> </Delete></pre>	Rows matching the WHERE clause specified by the FILTER element are deleted.	DELETE FROM [TABLE_NAME] WHERE WHERE_clause;
Delete Response	<pre><DeleteResponse xmlns="[VERSION]/InterfaceTables/[SCHEMA]/[APP_NAME]/[INTERFACETABLE_NAME]"> <DeleteResult>[rows deleted]</DeleteResult> </DeleteResponse></pre>	The number of rows deleted is returned in the DeleteResult element.	

Attribute descriptions:

[VERSION] = The message version string; for example, http://schemas.microsoft.com/OracleEBS/2008/05.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[APP_NAME] = Application short name.

[INTERFACETABLE_NAME] = Name of the interface table.

[FIELD1_NAME] = Table field name.

[COLUMN_list] = Comma-separated list of columns.

[WHERE_clause] = WHERE_clause for the SELECT statement used for the operation; for example, ID > 10.

◆ Important

The message structure for the basic operations on interface views, database tables, and database views is the same as that on interface tables, but the namespace for the operation specifies an interface view, database table, or database view rather than an interface table.

Message Actions for Basic Operations

The following table shows the message actions that the Oracle E-Business adapter uses for the basic operations on the interface tables and interface view in Oracle E-Business Suite, and the tables and views in the underlying database. The Oracle E-Business adapter uses the interface table, interface view, database table, or database view specified in the message action to determine the target of the operation.

📌 Note

See entity descriptions after table.

Operation	Message Action	Example
Insert	<p>Applications: InterfaceTables/Insert/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]</p> <p>Database: Tables/Insert/[SCHEMA]/[TABLE_NAME]</p>	<p>Applications: InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY</p> <p>Database: Tables/Insert/GL/GL_ALLOC_HISTORY</p>

Insert Response	<p>Applications: InterfaceTables/Insert/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]/response</p> <p>Database: Tables/Insert/[SCHEMA]/[TABLE_NAME]/response</p>	<p>Applications: InterfaceTables/Insert/SQLGL/GL/GL_ALLOC_HISTORY/response</p> <p>Database: Tables/Insert/GL/GL_ALLOC_HISTORY/response</p>
Select	<p>Applications: InterfaceTables/Select/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]</p> <p>Database: Tables/Select/[SCHEMA]/[TABLE_NAME]</p>	<p>Applications: InterfaceTables/Select/SQLGL/GL/GL_ALLOC_HISTORY</p> <p>Database: Tables/Select/GL/GL_ALLOC_HISTORY</p>
Select Response	<p>Applications: InterfaceTables/Select/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]/response</p> <p>Database: Tables/Select/[SCHEMA]/[TABLE_NAME]/response</p>	<p>Applications: InterfaceTables/Select/SQLGL/GL/GL_ALLOC_HISTORY/response</p> <p>Database: Tables/Select/GL/GL_ALLOC_HISTORY/response</p>
Update	<p>Applications: InterfaceTables/Update/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]</p> <p>Database: Tables/Update/[SCHEMA]/[TABLE_NAME]</p>	<p>Applications: InterfaceTables/Update/SQLGL/GL/GL_ALLOC_HISTORY</p> <p>Database: Tables/Update/GL/GL_ALLOC_HISTORY</p>
Update Response	<p>Applications: InterfaceTables/Update/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]/response</p> <p>Database: Tables/Update/[SCHEMA]/[TABLE_NAME]/response</p>	<p>Applications: InterfaceTables/Update/SQLGL/GL/GL_ALLOC_HISTORY/response</p> <p>Database: Tables/Update/GL/GL_ALLOC_HISTORY/response</p>
Delete	<p>Applications: InterfaceTables/Delete/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]</p> <p>Database: Tables/Delete/[SCHEMA]/[TABLE_NAME]</p>	<p>Applications: InterfaceTables/Delete/SQLGL/GL/GL_ALLOC_HISTORY</p> <p>Database: Tables/Delete/GL/GL_ALLOC_HISTORY</p>
Delete Response	<p>Applications: InterfaceTables/Delete/[SHORT_NAME]/[APP_NAME]/[TABLE_NAME]/response</p> <p>Database: Tables/Delete/[SCHEMA]/[TABLE_NAME]/response</p>	<p>Applications: InterfaceTables/Delete/SQLGL/GL/GL_ALLOC_HISTORY/response</p> <p>Database: Tables/Delete/GL/GL_ALLOC_HISTORY/response</p>

Entity descriptions:

- [SCHEMA] - Collection of Oracle artifacts (for example, GL).
- [TABLE_NAME] - Name of the table (for example, GL_ALLOC_HISTORY).

◆ Important

The message action for the Select operation on an interface view is the same as that for the interface table, except that "InterfaceViews" replaces "InterfaceTables." Similarly, the message action for the Select operation on a database view is the same as that for the database table, except that "Views" replaces "Tables."

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for Stored Procedures, Functions, and PL/SQL APIs

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces the underlying Oracle database stored procedures, functions, and PL/SQL APIs (stored procedures and functions within a package) as operations. This section describes the message structure and actions used to invoke stored procedures, functions, and PL/SQL APIs.

Message Structure of Stored Procedures, Functions, and PL/SQL APIs

The operations surfaced for functions and stored procedures follow a request-response message exchange pattern. The following table shows the structure of these request and response messages.

 Note
See entity descriptions after the table.

Operation	XML Message	Description
Stored Procedure Request	<pre><[SP_NAME] xmlns="[VERSION]/Procedures/[SCHEMA]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]></pre>	Supports Oracle IN and IN OUT parameters in the message body
Stored Procedure Response	<pre><[SP_NAME]Response xmlns="[VERSION]/Procedures/[SCHEMA]"> <[PRM1_NAME]>value1<[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]Response></pre>	Supports Oracle OUT and IN OUT parameters in the message body
Function Request	<pre><[FN_NAME] xmlns="[VERSION]/Functions/[SCHEMA]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[FN_NAME]></pre>	Supports Oracle IN and IN OUT parameters in the message body
Function Response	<pre><[FN_NAME]Response xmlns="[VERSION]/Functions/[SCHEMA]"> <[FN_NAME]Result>return_value</[FN_NAME]Result> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[FN_NAME]Response></pre>	Supports Oracle OUT and IN OUT parameters in the message body The function return value is returned in the <[FN_NAME]Result> element. This is the first element in the response message. It comes before any parameters.

PL/SQL API Request	<pre><[SP_NAME] xmlns="[VERSION]/PackageApis/[SCHEMA]/[PACKAGE_NAME]/[SP_NAME]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]></pre>	Same as Function or Stored Procedure
Packaged Procedure or Function Response	<pre><[SP_NAME]Response xmlns="[VERSION]/PackageApis/[SCHEMA]/[PACKAGE_NAME]/[SP_NAME]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]Response></pre>	Same as Function or Stored Procedure

Entity descriptions:

[VERSION] = <http://schemas.microsoft.com/OracleEBS/2008/05>.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[SP_NAME] = The stored procedure to be executed; for example, SP_INSERT.

[FN_NAME] = The function to be executed; for example, FN_GETID.

[PRM1_NAME] = The name of the Oracle parameter. See the Description column for supported parameter directions for each message.

[PACKAGE_NAME] = The name of the package that contains the targeted procedure or function.

The Oracle database supports overloading for stored procedures and functions. The Oracle E-Business adapter supports this capability by appending an overload string to the target namespace for each overloaded artifact. The value of this string is "overload1" for the first overload, "overload2" for the second overload, and so on. The following example shows the message structure for two overloaded stored procedures.

```
Stored Procedure Overload 1:
<[SP_NAME] xmlns="[VERSION]/PackageApis/[SCHEMA]/[PACKAGE_NAME]/[SP_NAME]/overload1">
  <[PRM1_NAME]>value1</[PRM1_NAME]>
  <[PRM2_NAME]>value1</[PRM2_NAME]>
  ...
</[SP_NAME]>

Stored Procedure Overload 2:
<[SP_NAME] xmlns="[VERSION]/PackageApis/[SCHEMA]/[PACKAGE_NAME]/[SP_NAME]/overload2">
  <[PRM1_NAME]>value1</I_[PRM1_NAME]>
  <[PRM2_NAME]>value1</I_[PRM2_NAME]>
  ...
</[SP_NAME]>
```

Message Actions of Stored Procedures, Functions, and PL/SQL APIs

The Oracle E-Business adapter uses the following message actions for stored procedure, function, and PL/SQL API operations.

Note
See entity descriptions after the table.

Message	Action	Example
---------	--------	---------

Stored Procedure Request	Procedures/[SCHEMA]/[SP_NAME]	Procedures/SCOTT/SP_INSERT
Stored Procedure Response	Procedures/[SCHEMA]/[SP_NAME]/response	Procedures/SCOTT/SP_INSERT/response
Function Request	Functions/[SCHEMA]/[FN_NAME]	Functions/SCOTT/FN_GETID
Function Response	Functions/[SCHEMA]/[FN_NAME]/response	Functions/SCOTT/FN_GETID/response
PL/SQL API Request	[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]	SCOTT/Package/CUSTOMER/SP_INSERT
Packaged Stored Procedure Response	[SCHEMA]/Package/[PACKAGE_NAME]/[SP_NAME]/response	SCOTT/Package/CUSTOMER/SP_INSERT/response
Packaged Function Request	[SCHEMA]/Package/[PACKAGE_NAME]/[FN_NAME]	SCOTT/Package/CUSTOMER/FN_GETID
Packaged Function Response	[SCHEMA]/Package/[PACKAGE_NAME]/[FN_NAME]/response	SCOTT/Package/CUSTOMER/FN_GETID/response
Overloaded Stored Procedure Request	[SCHEMA]/Procedure/[SP_NAME]/[OVERLOAD]	SCOTT/Procedure/SP_INSERT/overload1
Overloaded Stored Procedure Response	[SCHEMA]/Procedure/[SP_NAME]/[OVERLOAD]/response	SCOTT/Procedure/SP_INSERT/overload1/response

Entity descriptions:

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[SP_NAME] = The stored procedure to be executed; for example, SP_INSERT.

[FN_NAME] = The function to be executed; for example, FN_GETID.

[PACKAGE_NAME] = The name of the package that contains the targeted procedure or function.

[OVERLOAD] = The Overload parameter. The possible values are overload1, overload2, and so on.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for Concurrent Programs

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces concurrent programs as operations. Along with the concurrent programs exposed as operations, the Oracle E-Business adapter also surfaces the following three standard operations: Get_Status, Wait_For_Request, and Submit_Request. For information about these operations related to concurrent programs, see [Operations on Concurrent Programs](#).

Message Structure of Concurrent Program Operations

The operations surfaced for concurrent programs follow a request-response message exchange pattern. The following table shows the structure of these request and response messages.

 Note
See entity descriptions after the table.

Operation	XML Message	Description
-----------	-------------	-------------

[Concurrent_Program_Name]
Request

```
<?xml version="1.0" encoding="utf-8" ?>  
<[Concurrent_Program_Name] xmlns="[VERSION]"/>  
  <SetOptions>  
    <Implicit>[value]</Implicit>  
    <Protected>[value]</Protected>  
    <Language>[value]</Language>  
    <Territory>[value]</Territory>  
    <ContinueOnFail>[value]</ContinueOnFail>  
  </SetOptions>  
  <SetPrintOptions>  
    <Printer>[value]</Printer>  
    <Style>[value]</Style>  
    <Copies>[value]</Copies>  
    <SaveOutput>[value]</SaveOutput>  
    <PrintTogether>[value]</PrintTogether>  
    <ContinueOnFail>[value]</ContinueOnFail>  
  </SetPrintOptions>  
  <SetRepeatOptions>  
    <RepeatTime>[value]</RepeatTime>  
    <RepeatInterval>[value]</RepeatInterval>  
    <RepeatUnit>[value]</RepeatUnit>  
    <RepeatType>[value]</RepeatType>  
    <RepeatEndTime>[value]</RepeatEndTime>  
    <ContinueOnFail>[value]</ContinueOnFail>  
  </SetRepeatOptions>  
  <Description>[value]</Description>  
  <StartTime><[value]</StartTime>  
  <[CONCURRENT_PROGRAM_ARGUMENT1]>[value]</[CONCURRENT_PROGRAM_ARGUMENT1]>  
  <[CONCURRENT_PROGRAM_ARGUMENT2]>[value]</[CONCURRENT_PROGRAM_ARGUMENT2]>  
  ...  
</[Concurrent_Program_Name]>
```

- The [Concurrent_Program_Name] operation takes five standard parameters: *SetOptions*, *SetPrintOptions*, *SetRepeatOptions*, *Description*, and *StartTime*.
- The *ContinueOnFail* parameter indicates whether the concurrent request submission should continue in case the parent parameter (*SetOptions*, *SetPrintOptions*, or *SetRepeatOptions*) fails, or whether it should throw an exception. You can specify **True** (continue) or **False** (throw an exception).
- For detailed information about each parameter, see [Operations on Concurrent Programs](#).

<p>[Concurrent_Program_Name] Response</p>	<pre><?xml version="1.0" encoding="utf-8" ?> <[Concurrent_Program_Name]Response xmlns="[VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <[Concurrent_Program_Name]Result>[value] </[Concurrent_Program_Name]Result> </[Concurrent_Program_Name]Response></pre>	<p>The response from Oracle E-Business Suite contains a concurrent request ID.</p>
<p>Get_Status Request</p>	<pre><?xml version="1.0" encoding="utf-8" ?> <GetStatusForConcurrentProgram xmlns="[VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <RequestId>[value] </RequestId> </GetStatusForConcurrentProgram></pre>	<p>This Get_Status request message takes the request ID of a concurrent program as an input.</p>
<p>Get_Status Response</p>	<pre><?xml version="1.0" encoding="utf-8" ?> <GetStatusForConcurrentProgramResponse xmlns="[VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <GetStatusForConcurrentProgramResult>[value] </GetStatusForConcurrentProgramResult> <Phase>[value] </Phase> <Status>[value] </Status> <DevPhase>[value] </DevPhase> <DevStatus>[value] </DevStatus> <Message>[value] </Message> </GetStatusForConcurrentProgramResponse></pre>	<p>This Get_Status response message returns the request phase/status and the completion message of a concurrent program.</p> <p>For detailed information about each parameter, see Operations on Concurrent Programs.</p>

Wait_For_Request Request	<pre><?xml version="1.0" encoding="utf-8" ?> <WaitForRequestForConcurrentProgram xmlns=" [VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <RequestId>[value]</RequestId> <Interval>[value]</Interval> <MaxWait>[value]</MaxWait> </WaitForRequestForConcurrentProgram></pre>	For detailed information about each parameter, see Operations on Concurrent Programs .
Wait_For_Request Response	<pre><?xml version="1.0" encoding="utf-8" ?> <WaitForRequestForConcurrentProgramResponse xmlns=" [VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <WaitForRequestForConcurrentProgramResult>[value]</WaitForRequestForConcurrentProgramResult> <Phase>[value]</Phase> <Status>[value]</Status> <DevPhase>[value]</DevPhase> <DevStatus>[value]</DevStatus> <Message>[value]</Message> </WaitForRequestForConcurrentProgramResponse></pre>	This Wait_For_Request response message returns the request phase/status and the completion message of a concurrent program. For detailed information about each parameter, see Operations on Concurrent Programs .

Submit_Req
uest Re
quest

```
<?xml version="1.0" encoding="utf-8" ?>
<SubmitRequestForConcurrentProgram xmlns="
[VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]">
  <SetOptions>
    <Implicit>[value]
  </Implicit>
    <Protected>[value]
  </Protected>
    <Language>[value]
  </Language>
    <Territory>[value]
  </Territory>
    <ContinueOnFail>[value]
  </ContinueOnFail>
  </SetOptions>
    <SetPrintOptions>
      <Printer>[value]
    </Printer>
      <Style>[value]
    </Style>
      <Copies>[value]
    </Copies>
      <SaveOutput>[value]
    </SaveOutput>
      <PrintTogether>[value]
    </PrintTogether>
      <ContinueOnFail>[value]
    </ContinueOnFail>
    </SetPrintOptions>
    <SetRepeatOptions>
      <RepeatTime>[value]
    </RepeatTime>
      <RepeatInterval>[value]
    </RepeatInterval>
      <RepeatUnit>[value]
    </RepeatUnit>
      <RepeatType>[value]
    </RepeatType>
      <RepeatEndTime>[value]
    </RepeatEndTime>
      <ContinueOnFail>[value]
    </ContinueOnFail>
    </SetRepeatOptions>

    <Program>[value]
  </Program>
    <Description>[value]
  </Description>
    <StartTime>[value]
  </StartTime>
    <Arguments>[array_of_strings]
  </Arguments>
</SubmitRequestForConcurrentProgram>
```

For detailed information about each parameter, see [Operations on Concurrent Programs](#).

Submit_Request Response	<pre><?xml version="1.0" encoding="utf-8" ?> <SubmitRequestForConcurrentProgramResponse xmlns="[VERSION]/ConcurrentPrograms/[APP_SHORT_NAME]"> <SubmitRequestForConcurrentProgramResult>[value]</SubmitRequestForConcurrentProgramResult> </SubmitRequestForConcurrentProgramResponse></pre>	If the submit request completes successfully, the response message returns the concurrent request ID. Otherwise, it returns "0".
-------------------------	--	--

Entity descriptions:

[VERSION] = <http://schemas.microsoft.com/OracleEBS/2008/05>

[APP_SHORT_NAME] = Application short name

[CONCURRENT_PROGRAM_ARGUMENT] = Argument expected by the concurrent program as defined in Oracle E-Business Suite

Message Actions for Concurrent Programs

The Oracle E-Business adapter uses the following message actions for concurrent programs.

Note
See entity descriptions after the table.

Message	Action	Example
[Concurrent_Program_Name] Request	ConcurrentPrograms/[APP_SHORT_NAME]/[CONCURRENT_PROGRAM_SHORT_NAME]	ConcurrentPrograms/SQLGL/ADSFINS
[Concurrent_Program_Name] Response	ConcurrentPrograms/[APP_SHORT_NAME]/[CONCURRENT_PROGRAM_SHORT_NAME]/response	ConcurrentPrograms/SQLGL/ADSFINS/response
Get_Status Request	ConcurrentPrograms/[APP_SHORT_NAME]/GetStatusForConcurrentProgram	ConcurrentPrograms/SQLGL/GetStatusForConcurrentProgram
Get_Status Response	ConcurrentPrograms/[APP_SHORT_NAME]/GetStatusForConcurrentProgram/response	ConcurrentPrograms/SQLGL/GetStatusForConcurrentProgram/response
Wait_For_Request Request	ConcurrentPrograms/[APP_SHORT_NAME]/WaitForRequestForConcurrentProgram	ConcurrentPrograms/SQLGL/WaitForRequestForConcurrentProgram
Wait_For_Request Response	ConcurrentPrograms/[APP_SHORT_NAME]/WaitForRequestForConcurrentProgram/response	ConcurrentPrograms/SQLGL/WaitForRequestForConcurrentProgram/response
Submit_Request Request	ConcurrentPrograms/[APP_SHORT_NAME]/SubmitRequestForConcurrentProgram	ConcurrentPrograms/SQLGL/SubmitRequestForConcurrentProgram
Submit_Request Response	ConcurrentPrograms/[APP_SHORT_NAME]/SubmitRequestForConcurrentProgram/response	ConcurrentPrograms/SQLGL/SubmitRequestForConcurrentProgram/response

Entity descriptions:

[APP_SHORT_NAME] = Application short name

[CONCURRENT_PROGRAM_SHORT_NAME] = Concurrent Program short name

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for Request Sets

Each request set in an Oracle application in Oracle E-Business Suite is surfaced as an operation in Microsoft BizTalk Adapter for Oracle E-Business Suite.

Message Structure of Request Set Operation

The operations surfaced for request set follow a request-response message exchange pattern. The following table shows the structure of these request and response messages.

 Note
See entity descriptions after the table.

Operation	XML Message	Description
[Request_Set_Name] Request	<pre><?xml version="1.0" encoding="utf-8" ?> <[Request_Set_Name] xmlns="[VERSION]/RequestSets/[APP_SHORT_NAME]/"> <SetRelClassOptions> <Application> [value]</Application> <ClassName>[value]</ClassName> <CancelOrHold>[value]</CancelOrHold> <StaleDate>[value]</StaleDate> <ContinueOnFail>[value]</ContinueOnFail> </SetRelClassOptions> <SetPrintOptions> <Printer>[value]</Printer> <Style>[value]</Style> <Copies>[value]</Copies> <SaveOutput>[value]</SaveOutput> <PrintTogether>[value]</PrintTogether> <ContinueOnFail>[value]</ContinueOnFail> </SetPrintOptions> <SetRepeatOptions> <RepeatTime>[value]</RepeatTime> <RepeatInterv</pre>	<ul style="list-style-type: none"> • The [Request_Set_Name] operation takes five standard parameters: <i>SetRelClassOptions</i>, <i>SetPrintOptions</i>, <i>SetRepeatOptions</i>, <i>SetNlsOptions</i>, and <i>StartTime</i>. • The <i>ContinueOnFail</i> parameter indicates whether the request set submission should continue or throw an exception in case the parent parameter (<i>SetRelClassOptions</i>, <i>SetPrintOptions</i>, <i>SetRepeatOptions</i> or <i>SetNlsOptions</i>) fails. You can specify True (continue) or False (throw an exception). • For detailed information about each parameter, see Operations on Request Sets.

	<pre> al>[value]</RepeatInterval> <RepeatUnit>[value]</RepeatUnit> <RepeatType>[value]</RepeatType> <RepeatEndTime>[value]</RepeatEndTime> <ContinueOnFail>[value]</ContinueOnFail> </SetRepeatOptions> <SetNlsOptions> <Language>[value]</Language> <Territory>[value]</Territory> <ContinueOnFail>[value]</ContinueOnFail> </SetNlsOptions> <StartTime><[value]</StartTime> <[CONCURRENT_PROGRAM_NAME1]>[value]</[CONCURRENT_PROGRAM_NAME1]> <[CONCURRENT_PROGRAM_NAME2]>[value]</[CONCURRENT_PROGRAM_NAME2]> ... </[Request_Set_Name]> </pre>	
[Request_Set_Name] Response	<pre> <?xml version="1.0" encoding="utf-8" ?> <[Request_Set_Name]Response xmlns=" [VERSION]/RequestSets/[APP_SHORT_NAME]"> <[Request_Set_Name]Result>[value] </[Request_Set_Name]Result> </[Request_Set_Name]Response> </pre>	The response from Oracle E-Business Suite contains a concurrent request ID.

Entity descriptions:

[VERSION] = <http://schemas.microsoft.com/OracleEBS/2008/05>

[CONCURRENT_PROGRAM_NAME] = Concurrent program included in the request set.

Message Actions for Request Sets

The Oracle E-Business adapter uses the following message actions for request sets.

Note
See entity descriptions after the table.

Message	Action	Example
Request Set request	RequestSets/[APP_SHORT_NAME]/[REQUESTSET_SHORT_NAME]	RequestSets/SQLGL/FNDRSSUB965
Request Set response	RequestSets/[APP_SHORT_NAME]/[REQUESTSET_SHORT_NAME]/response	RequestSets/SQLGL/FNDRSSUB965/response

Entity descriptions:

[APP_SHORT_NAME] = Application short name.

[REQUESTSET_SHORT_NAME] = Request Set short name.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for Special LOB Operations

The Read_<LOBColName> and Update_<LOBColName> operations are surfaced for tables and views that contain LOB columns, where <LOBColName> is the LOB column in the table or view. These operations enable you to read or write the LOB data as a stream of base64Binary-encoded data. They operate on a single column of LOB data in a single row.

For an overview of the Read_<LOBColName> and Update_<LOBColName> operations and of the Oracle LOB data types supported, see [Operations on Interface Tables, Interface Views, Tables, and Views That Contain LOB Data](#).

Message Structure of LOB Data-Type Operations

The following table shows the structure of the request and response messages for the Read_<LOBColName> and Update_<LOBColName> operations. The target table for the operation is specified in the message action and also appears in the target namespace.

Note
See entity descriptions after the table.

Operation	XML Message	Description
Read_<LOBColName>	<pre><Read_[LOBColName] xmlns="[VERSION]/Tables/[SCHEMA]/[TABLE_NAME]"> <FILTER>[WHERE_clause]</FILTER> </Read_[LOBColName]></pre>	The LOB data in the row that matches the where clause specified in the FILTER element is returned. The where clause should match only a single row. If there is more than one matching row, the Oracle E-Business adapter will throw an exception.
Read_<LOBColName> Response	<pre><Read_[LOBColName]Response xmlns="[VERSION]/Tables/[SCHEMA]/[TABLE_NAME]"> <Read_[LOBColName]Result> [LOB_DATA] </Read_[LOBColName]Result> </Read_[LOBColName]Response></pre>	The LOB data is returned as a stream of base64Binary encoded data.
Update_<LOBColName>	<pre><Update_[LOBColName] xmlns="[VERSION]/Tables/[SCHEMA]/[TABLE_NAME]"> <FILTER>[WHERE_clause]</LOB_COLUMN> <DATA>[Value]</DATA> </Update_[LOBColName]></pre>	<p>The LOB data in the row that matches the where clause specified in the FILTER element is updated with the data in the <DATA> element. The where clause should match only a single row. If there is more than one matching row, the Oracle E-Business adapter throws an exception.</p> <p>Note While updating BLOB columns, the <DATA> element must always contain a base64 encoded value. For CLOB and NCLOB, the <DATA> element can have string values.</p>

Update_<LOBColName> Response	<pre><Update_[LOBColName]Response xmlns="[VERSION]/Tables/[SCHEMA]/[TABLE_NAME]"> </Update_[LOBColName]Response></pre>	An empty response is returned.
------------------------------	--	--------------------------------

Entity descriptions:

[VERSION] = The message version string; for example, "http://schemas.microsoft.com/OracleEBS/2008/05".

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = The table that contains the targeted LOB column; for example, CUSTOMER.

[LOBCol_Name] = The name of a LOB column; for example, Photo.

[WHERE_clause] = An Oracle database SELECT statement WHERE clause that matches a single row; for example, ID = 1.

[LOB_DATA] = The LOB column data in base64Binary type.

<p>Important</p>
<p>The message structure for the Read_<LOBColName> and Update_<LOBColName> operations on views is the same as that on tables except that the namespace for the operation specifies a view rather than a table: <ReadLOB xmlns = "[VERSION]/Views/[SCHEMA]/[VIEW_NAME]">.</p>

Message Actions for LOB Data-Type Operations

The following table shows the message actions that are used by the Oracle E-Business adapter for the Read_<LOBColName> and Update_<LOBColName> operations on tables. The Oracle E-Business adapter uses the table name and the LOB column name specified in the message action to determine the target table and LOB column for the operation.

<p>Note</p>
<p>See entity descriptions after the table.</p>

Operation	Action	Example
Read_<LOBColName>	Tables/ReadLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]	Tables/ReadLOB/SCOTT/CUSTOMER/Photo
Read_<LOBColName> Response	Tables/ReadLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]/response	Tables/ReadLOB/SCOTT/CUSTOMER/Photo/response
Update_<LOBColName>	<p>For BLOB:</p> <p>Tables/UpdateBLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]</p> <p>For CLOB and NCLOB:</p> <p>Tables/UpdateCLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]</p>	<p>For BLOB:</p> <p>Tables/UpdateBLOB/SCOTT/CUSTOMER/Photo/</p> <p>For CLOB and NCLOB:</p> <p>Tables/UpdateCLOB/SCOTT/CUSTOMER/Photo1/</p>
Update_<LOBColName> Response	<p>For BLOB:</p> <p>Tables/UpdateBLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]/response</p> <p>For CLOB and NCLOB:</p> <p>Tables/UpdateCLOB/[SCHEMA]/[TABLE_NAME]/[LOBColName]/response</p>	<p>For BLOB:</p> <p>Tables/UpdateBLOB/SCOTT/CUSTOMER/Photo/response</p> <p>For CLOB and NCLOB:</p> <p>Tables/UpdateCLOB/SCOTT/CUSTOMER/Photo1/response</p>

Entity descriptions:

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = The table that contains the targeted LOB column; for example, CUSTOMER. (The SCOTT.CUSTOMER table is installed by a SQL script included in the samples.)

[LOBCol_Name] = The name of a LOB column; for example, Photo.

◆ Important

The message action for Read_<LOBColName> and Update_<LOBColName> operations on views is similar to that used for tables, except that action for the operation specifies a view rather than a table: Views/ReadLOB/[SCHEMA]/[VIEW_NAME]/[LOBColName].

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for the Polling Operations

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces various inbound operations related to polling depending on the target object in Oracle E-Business Suite. For interface tables, interface views, tables, and views, a single Poll operation is surfaced whereas you can have multiple custom polling operations for PL/SQL APIs, functions, and stored procedures.

You configure the polling operations by setting binding properties in the Oracle E-Business adapter. For more information about these binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). You set the **PollingStatement** binding property to specify a SQL statement, stored procedure, function or a procedure within a package for the polling query. The result set of this query is returned as data to your code in the polling operation.

Message Structure for the Polling Operations

The following table shows the XML message structure for the various polling operations.

Note
See entity descriptions after the table.

Operation	Target Object	XML Message	Description
Poll	<ul style="list-style-type: none"> Interface Tables Interface Views Tables Views 	<pre><?xml version="1.0" encoding="utf-8" ?> <Poll xmlns="[Version]/[TargetObject]/[Schema]/[TargetObjectName]"> <DATA> <SelectRecord> <Column1>[Value]</Column1> <Column2>[Value]</Column2> ... </SelectRecord> </DATA> </Poll></pre>	<p>For example, the XML message for the Poll operation on Interface Tables will be as follows:</p> <pre><?xml version="1.0" encoding="utf-8" ?> <Poll xmlns="[Version]/InterfaceTables/[Schema]/[InterfaceTableName]"> <DATA> <SelectRecord> <Column1>[Value]</Column1> <Column2>[Value]</Column2> ... </SelectRecord> </DATA> </Poll></pre>

<p>[CustomPollingOperation]</p>	<ul style="list-style-type: none"> • PL/SQL APIs • Stored Procedures • Functions 	<p>PL/SQL APIs</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/PollingPackageAPis/[Schema]/[PL/SQL API]"> <[CustomPollingOperation]Result>[Value]</[CustomPollingOperation]Result> </[CustomPollingOperation]></pre> <p>Functions</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/PollingFunctions/[Schema]"> <[CustomPollingOperation]Result> <COL1>[Value]</COL1> <COL2>[Value]</COL2> ... </[CustomPollingOperation]Result> </[CustomPollingOperation]></pre> <p>Stored Procedures</p> <pre><?xml version="1.0" encoding="utf-8" ?> <[CustomPollingOperation] xmlns="[Version]/PollingFunctions/[Schema]"> <[CustomPollingOperation]Result> <PRM1>[Value]</PRM1> <PRM2>[Value]</PRM2> ... </[CustomPollingOperation]Result> </[CustomPollingOperation]></pre>	<p>The structure of the result set in the polling operation is determined by the data type of the elements in the target object.</p>
---------------------------------	---	---	--

Entity descriptions:

[Version] = <http://schemas.microsoft.com/OracleEBS/2008/05>.

[CustomPollingOperation] = Name of the custom polling operation.

[Schema] = Name of the Oracle schema. For example, SCOTT.

[PL/SQL API] = Name of the PL/SQL API on which a custom polling operation is performed.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for the Notification Operation

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces the Notification operation to receive database change notifications from the underlying database in Oracle E-Business Suite.

You configure the Notification operation by setting binding properties in the Oracle E-Business adapter. For more information about the Notification-related binding properties, see [Working with BizTalk Adapter for Oracle E-Business Suite Binding Properties](#). You set the **NotificationStatement** binding property to specify a SELECT statement for the query notification.

Message Structure for the Notification Operation

The following table shows the XML message structure for the Notification operation.

Operation	XML Message	Description
Notification	<pre><?xml version="1.0" encoding="utf-8" ?> <Notification xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Notification"> <Info>Value</Info> <Source>Value</Source> </Notification> <Type>Value</Type> </Notification></pre>	<p>This is the inbound message that is sent by Oracle E-Business Suite to the adapter clients. In the message:</p> <ul style="list-style-type: none"> • The <code><Info></code> tag indicates the reason for the notification. For example, an "insert" value in this tag indicates that data has been inserted in one or more of the tables referenced in the notification statement. • The <code><Source></code> tag indicates the source for the notification. For example, a "data" value in this tag indicates a change in the data in a referenced object. Similarly, an "object" value in this tag indicates a change in a referenced object. • The <code><Type></code> tag indicates the type of data change. For example, an "Update" value in the <code><Type></code> tag indicates that the results of the query have been updated.

Message Action for the Notification Operation

The message action for the notification operation is "Notification."

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Message Schemas for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The Microsoft BizTalk Adapter for Oracle E-Business Suite exposes the ExecuteNonQuery, ExecuteReader, and ExecuteScalar outbound operations at the root level to execute any arbitrary SQL statements or PL/SQL blocks in Oracle E-Business Suite.

For more information about:

- These operations, see [Support for ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).
- Performing these operations using the Oracle E-Business adapter, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations Using BizTalk Server](#).

Message Structure for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The messages in these operations follow a request-response message exchange pattern, and the following table shows the structure of these request and response messages.

Note
See entity descriptions after the table.

Operation	XML Message
ExecuteNonQuery Request	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteNonQuery xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <Query>[PL/SQL block]</Query> <OutputRefCursorNames> <string>[stringvalue1]</string> <string>[stringvalue2]</string> ... </OutputRefCursorNames> </ExecuteNonQuery></pre>
ExecuteNonQuery Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteNonQueryResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <ExecuteNonQueryResult>[value]</ExecuteNonQueryResult> <OutputRefCursors> <DataSet> <Any>[value]</Any> <Any>[value]</Any> ... </DataSet> </OutputRefCursors> </ExecuteNonQueryResponse></pre>
ExecuteReader Request	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteReader xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <Query>[PL/SQL block]</Query> </ExecuteReader></pre>

ExecuteReader Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteReaderResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <ExecuteReaderResult> <Any>[value]</Any> <Any>[value]</Any> ... </ExecuteReaderResult> </ExecuteReaderResponse></pre>
ExecuteScalar Request	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteScalar xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <Query>[PL/SQL block]</Query> </ExecuteScalar></pre>
ExecuteScalar Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteScalarResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/GenericOperation/ "> <ExecuteScalarResult>[value]</ExecuteScalarResult> </ExecuteScalarResponse></pre>

Entity descriptions:

[PL/SQL block] = The entire PL/SQL block to be executed.

[stringvalue1] = A value in the array of strings.

Message Action for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The following table shows the message actions that are used by the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations.

Operation	Action
ExecuteNonQuery Request	GenericOp/ExecuteNonQuery
ExecuteNonQuery Response	GenericOp/ExecuteNonQuery/response
ExecuteReader Request	GenericOp/ExecuteReader
ExecuteReader Response	GenericOp/ExecuteReader/response
ExecuteScalar Request	GenericOp/ExecuteScalar
ExecuteScalar Response	GenericOp/ExecuteScalar/response

Message Schemas for the Composite Operation

The Microsoft BizTalk Adapter for Oracle E-Business Suite enables you to execute composite operations in Oracle E-Business Suite. A composite operation can contain multiple operations, and in any order. For information about which operations can be included in a composite operation, see [Support for Composite Operations](#).

For information about how to perform composite operations using the Oracle E-Business adapter, see [Performing Composite Operations on Oracle Database by Using BizTalk Server](#).

Message Structure for the Composite Operation

Since a composite operation contains multiple individual operations; the message structure of a composite operation contains message structures of the individual operations. The composite operation message follows a request-response message exchange pattern.

The following table shows the structure of the request and response messages of a composite operation that contains an Insert operation, a packaged stored procedure that does not take any input parameters, and a Delete operation.

Note
See entity descriptions after the table.

Operation	XML Message
Composite Operation Request	<pre> <?xml version="1.0" encoding="utf-8" ?> <Request xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <Insert xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/[SCHEMA]/[TABLE_NAME]"> <Recordset> <InsertRecord> <[FIELD1_NAME]>[value1]</[FIELD1_NAME]> <InlineValue>[value]</InlineValue> <[FIELD2_NAME]>[value2]</[FIELD2_NAME]> <InlineValue>[value]</InlineValue> ... <InsertRecord> </RECORDSET> </Insert> <[SP_NAME] xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/PackageApis/[SCHEMA]/[APP_NAME]" /> <Delete xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/Tables/[SCHEMA]/[TABLE_NAME]"> <FILTER>[WHERE_clause]</FILTER> </Delete> </Request> </pre>

Composite Operation Response	<pre> <?xml version="1.0" encoding="utf-8" ?> <RequestResponse xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <InsertResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/T ables/[SCHEMA]/[TABLE_NAME]"> <InsertResult>[value]</InsertResult> </InsertResponse> <[SP_NAME]Response xmlns="http://schemas.microsoft.com/OracleEBS/2008/0 5/Procedures/[SCHEMA]"> <[PRM1_NAME]>value1<[PRM1_NAME]> <[PRM2_NAME]>value2<[PRM2_NAME]> ... </[SP_NAME]Response> <DeleteResponse xmlns="http://schemas.microsoft.com/OracleEBS/2008/05/T ableOp/[SCHEMA]/[TABLE_NAME]"> <DeleteResult>[value]</DeleteResult> </DeleteResponse> </RequestResponse> </pre>
------------------------------	---

Entity descriptions:

[PROJECT_NAME] = Name of the BizTalk project that contains the composite operation schema.

[COMPOSITE_SCHEMA_NAME] = Name of the composite operation schema given by the user.

[SCHEMA] = Collection of Oracle artifacts; for example, SCOTT.

[TABLE_NAME] = Name of the table; for example, EMPLOYEE.

[FIELD1_NAME] = Table field name; for example, NAME.

[SP_NAME] = The packaged stored procedure to be executed; for example, ADD_EMP_DETAILS.

[APP_NAME] = Name of the Oracle Application that contains the packaged stored procedure.

[PRM1_NAME] = The name of the Oracle parameter in the packaged stored procedure.

Message Action for the Composite Operation

The message action for the composite operation is "CompositeOperation."

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Oracle E-Business Suite](#)

Glossary

The following terms and definitions are used in Microsoft BizTalk Adapter for Oracle E-Business Suite Help.

Microsoft BizTalk Adapter Pack Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

Term	Definition
adapter	A WCF-based component that helps exchange messages between applications (for example, a line-of-business system) and BizTalk Server. The adapter consists of design-time components and run-time components for receive and send operations.
adapter client	An application that interacts with a line-of-business (LOB) system through the adapter.

B

Term	Definition
binding	A process by which software components and layers are linked together. When a network component is installed, the binding relationships and dependencies for the components are established. Binding allows components to communicate with each other. In BizTalk Server, an established mapping between an orchestration adapter-agnostic endpoint (port or role link) and physical adapter-specific endpoints (send/receive ports or party).
BizTalk Server	Connects diverse software. BizTalk Server enables you to create and modify process logic that uses that software. BizTalk Server also enables information workers to monitor running processes, interact with trading partners, and perform other business-oriented tasks.

C

Term	Definition
channel	A concrete implementation of a binding element. The binding represents the configuration, and the channel is the implementation associated with that configuration. Therefore, there is a channel associated with each binding element. Channels stack on top of each other to create the concrete implementation of the binding: the channel stack.
connection URL	A string that identifies a resource in a distributed environment. Adapters use a connection Uniform Resource Identifier (URI) that contains the information necessary to establish a connection with the LOB system.
contract	Specifies the collection and structure of messages required to access the operations offered by the service.

D

Term	Definition
data manipulation language (DML)	The subset of SQL statements that is used to retrieve and manipulate data. DML statements typically start with SELECT, INSERT, UPDATE, or DELETE.
design-time experience	Procedures and operations that a developer performs during design time; for example, using the Consumer Adapter Service BizTalk Project Add-in to retrieve message schemas.

E

Term	Definition
------	------------

endpoint address	A network address that identifies the location of a Windows Communication Foundation (WCF) service endpoint. For an adapter, the endpoint address is expressed as a connection Uniform Resource Identifier (URI) that contains location and connection parameters. The adapter can use these to establish a connection to the underlying line-of-business (LOB) system.
Enterprise Single Sign-on system	An SSO database, a master secret server, and one or more Enterprise Single Sign-On (SSO) servers. These servers do the mapping between the Windows and non-Windows credentials, look up the credentials in the SSO database, and are used for administering the SSO system. The SSO database is also used as a configuration store to hold custom configuration data for adapters.
Extensible Markup Language	A markup language designed to describe data. XML tags are not predefined.

G

Term	Definition
GAC	See global assembly cache.
global assembly cache (GAC)	A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer. Applications deployed in the global assembly cache must have a strong name.

H

Term	Definition
Health and Activity Tracking (HAT)	A user interface that enables specified data to be tracked. This interface can be used to monitor the health of the BizTalk Server farm and to track and view specific events and messages.

I

Term	Definition
inbound operation	An operation that is invoked by a line-of-business (LOB) system on the adapter.

M

Term	Definition
metadata	In WCF, refers to a description of the contract exposed by a service. This is known as the service description and is expressed in a WSDL document. The metadata exposed by an adapter describes the (interface to) the operations that it can perform on the underlying LOB system.
Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK	The specifications for building BizTalk adapters using open standards based on Web services.

O

Term	Definition
one-way	A message exchange pattern (MEP) in which the sender sends a message, but no response is returned by the receiver. In BizTalk Server, MEPs are referred to as communication patterns.
outbound operation	An operation that is invoked by the adapter on the line-of-business system (LOB).
output.cs	The default output file created by the ServiceModel Metadata Utility tool (svcutil.exe).

P

Term	Definition
polling	A technique that device drivers use to find out from multiple devices whether they contain data to transmit. The devices are polled one at a time.

proxy	In WCF, refers to a managed-code object that implements the service contract exposed by a service. The WCF service model is based on the use of such proxies. In the WCF service model, the service contract is expressed as a .NET interface.
-------	--

R

Term	Definition
request-reply	A message exchange pattern (MEP) in which the sender sends a request message and expects a response message from the receiver. In BizTalk Server, MEPs are referred to as communication patterns. Depending on the messaging technology and the direction of the request message (inbound or outbound), this pattern is also called request-reply or solicit-response.
runtime experience	Procedures and operations performed by a developer during run time or when deploying a solution; for example, creating a physical port binding from the BizTalk Server Administration console.

S

Term	Definition
schema	The structure for a message. A schema can contain multiple subschema.
ServiceModel Metadata Utility Tool (svcutil.exe)	A command-line utility that is included with WCF. It is used to create service model proxy code from the service description (metadata) that is exposed by a WCF service such as an adapter. For outbound operations, the tool creates a WCF client class and helper code; for inbound operations, the tool creates a WCF service contract and helper code.
Simple Object Access Protocol	See SOAP.
SOAP	A simple, XML-based protocol for exchanging structured and type information in decentralized, distributed environments. WCF is based on the exchange of SOAP messages between clients and services to invoke operations and return results.
SOAP message	A well-formed XML document. It should use the SOAP envelope and SOAP encoding namespaces and include an optional XML declaration, followed by a SOAP envelope (the root element), which is made up of an optional SOAP header and a SOAP message body.
SQL Server Integration Services (SSIS)	A component that is used to import, export, and transform data from different data sources. Previously called data transformation service (DTS).
SSIS	See SQL Server Integration Services.
SSO	See Enterprise Single Sign-on system.
strongly-typed data	A data set or result set that is bound to an underlying object type. Each row in a strongly-typed XML data set is composed of typed, named elements that correspond to fields of the underlying object type.
svcutil.exe	See ServiceModel Metadata Utility Tool.

U

Term	Definition
Uniform Resource Identifier (URI)	See connection URI

W

Term	Definition
------	------------

WCF	See Windows Communication Foundation
WCF channel model	A programming model that relies on several interfaces and other types. Channels provide a low-level programming model for sending and receiving messages.
WCF client	A client-application construct that exposes the service operations as methods. You can use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool to generate a WCF client class from the metadata exposed by an adapter.
WCF LOB Adapter SDK	See Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
WCF service contract	A managed-code representation of the service contract. It is expressed as an interface in which classes and methods are attributed to define the service, operation, message, and data contracts used to communicate with a service. You can use the ServiceModel Metadata Utility tool or the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF service contract from the metadata exposed by an adapter. You implement the WCF service contract to receive operations from an LOB system.
WCF service model	A WCF programming model in which a service is represented as a managed code object. The operations exposed by the service are represented as methods with strongly-typed data.
weakly-typed data	A data set or result set that is not bound to an underlying object type. Each row in a weakly-typed XML data set is composed of a collection of generic columns in which attributes describe the name and type of each element.
Web services	A unit of application logic providing data and services to other applications. Applications access XML Web services using standard Web protocols and data formats such as HTTP, XML, and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web , and are a cornerstone of the Microsoft .NET programming model.
Web Services Description Language (WSDL)	An XML-based language that describes a service as a set of endpoints that operate on messages. The WSDL document describes the service contract, operation contracts, message contracts, and data contracts that a client must use to interface with the service.
Windows Communication Foundation (WCF)	A Microsoft service-oriented communication infrastructure. The framework inherently provides clients with a service programming model and a channel programming model for finer control of message exchanges.
WSDL	See Web Services Description Language.

X

Term	Definition
XML	See Extensible Markup Language.
XML Schema definition language (XSD)	A schema language. An XML Schema defines the elements, attributes, and data types that comply with the World Wide Web Consortium (W3C) XML Schema Part 1: Structures Recommendation for the XML Schema Definition Language. The W3C XML Schema Part 2: Datatypes Recommendation is the recommendation for defining data types that are used in XML schemas. The XML Schema definition language enables you to define the structure and data types for XML messages.
XSD	See XML Schema definition language.

Class Library

Namespaces

Namespace
Microsoft.Adapters.OracleEBS
Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Microsoft.Adapters.OracleEBS Namespace

Classes

Class	Description
MlsSettings	Multiple Language System Settings configuration element
OracleEBSAdapter	The Binding Element for the Oracle E-Business Suite Adapter.
OracleEBSBinding	Represents the Oracle E-Business Suite WCF Binding.
OracleEBSBindingCollectionElement	Represents the StandardBindingCollectionElement for the Oracle E-Business Suite Adapter.
OracleEBSBindingConfigurationElement	Represents the StandardBindingElement for the Oracle E-Business Suite Adapter.
OracleEBSBindingElementExtensionElement	Enables the configuration of the Oracle E-Business Suite Adapter Binding from an application configuration file.
OracleEBSConnectionUri	Builds the Uniform Resource Identifier (URI) for establishing connection to the Oracle E-Business Suite and essentially the underlying Oracle database.
OracleEBSInboundTransactionBehavior	

Enumerations

Enumeration	Description
ClientCredential	The set of credentials that are specified while establishing a connection with the Oracle E-Business Suite.
InboundOperation	The Inbound Operation to be performed.
OracleServiceType	

ClientCredential Enumeration

The set of credentials that are specified while establishing a connection with the Oracle E-Business Suite.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
Public Enumeration ClientCredential
```

C#

```
public enum ClientCredential
```

C++

```
public enum class ClientCredential
```

J#

```
public enum ClientCredential
```

JScript

```
public enum ClientCredential
```

Members

Member name	Description
Database	The Database credentials.
EBusiness	The E-Business Suite credentials.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleEBS Namespace](#)

InboundOperation Enumeration

The Inbound Operation to be performed.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Enumeration InboundOperation
```

C#

```
public enum InboundOperation
```

C++

```
public enum class InboundOperation
```

J#

```
public enum InboundOperation
```

JScript

```
public enum InboundOperation
```

Members

Member name	Description
Notification	The Notification Inbound Operation.
Polling	The Polling Inbound Operation.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Class

Multiple Language System Settings configuration element

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Class MlsSettings
    Inherits ConfigurationElement
```

C#

```
public class MlsSettings : ConfigurationElement
```

C++

```
public ref class MlsSettings : public ConfigurationElement
```

J#

```
public class MlsSettings extends ConfigurationElement
```

JScript

```
public class MlsSettings extends ConfigurationElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

Microsoft.Adapters.OracleEBS.MlsSettings

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Members

Multiple Language System Settings configuration element

The following tables list the members exposed by the [MlsSettings](#) type.

Public Constructors

Name	Description
MlsSettings	Overloaded. Initializes a new instance of the MlsSettings class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 Calendar	Gets or sets the calendar used in the management license settings.
 Comparison	Gets or sets the comparison used in the management license settings.
 Currency	Gets or sets the currency used in the management license settings.
 DateFormat	Gets or sets the date format used in the used in the management license settings.
 DateLanguage	Gets or sets the date format used in the management license settings.
 DualCurrency	Gets or sets the dual currency used in the management license settings.
 ElementInformation	(inherited from ConfigurationElement)
 ISOCurrency	Gets or sets the ISO currency used in the management license settings.
 Language	Gets or sets the language used in the management license settings.
 LengthSemantics	Gets or sets the length semantics used in the management license settings.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 NCharConversionException	Gets or sets whether the specified object is a national character conversion exception.
 NumericCharacters	Gets or sets the numeric characters used in the management license settings.
 Sort	Gets or sets the sort order used in the management license settings.
 Territory	Gets or sets the territory used in the management license settings.
 TimeStampFormat	Gets or sets the time stamp format used in the management license settings.
 TimeStampTZFormat	Gets or sets the time stamp time zone format used in the management license settings.
 TimeZone	Gets or sets time zone used in the management license settings.

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	Overridden. Gets the properties used in the management license settings.

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	Overridden. Creates a string representation of the current MlsSettings configuration element.

Top

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)

 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ConfigurationElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[MIsSettings Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Constructor

Initializes a new instance of the [MlsSettings](#) class.

Overload List

Name	Description
MlsSettings ()	Initializes a new instance of the MlsSettings configuration element
MlsSettings (MlsSettings)	Initializes a new instance of the MlsSettings configuration element from an existing instance.

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Constructor ()

Initializes a new instance of the MlsSettings configuration element

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public MlsSettings ()
```

C++

```
public:  
MlsSettings ()
```

J#

```
public MlsSettings ()
```

JScript

```
public function MlsSettings ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Constructor (MlsSettings)

Initializes a new instance of the MlsSettings configuration element from an existing instance.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")> _
<SuppressMessageAttribute("Microsoft.Design", "CA1062:ValidateArgumentsOfPublicMethods")> _
Public Sub New ( _
    other As MlsSettings _
)
```

C#

```
[SuppressMessageAttribute("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
[SuppressMessageAttribute("Microsoft.Design", "CA1062:ValidateArgumentsOfPublicMethods")]
public MlsSettings (
    MlsSettings other
)
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Usage", L"CA2214:DoNotCallOverridableMethodsInConstructors")]
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1062:ValidateArgumentsOfPublicMethods")]
public:
MlsSettings (
    MlsSettings^ other
)
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors") */
/** @attribute SuppressMessageAttribute("Microsoft.Design", "CA1062:ValidateArgumentsOfPublicMethods") */
public MlsSettings (
    MlsSettings other
)
```

JScript

```
SuppressMessageAttribute("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")
SuppressMessageAttribute("Microsoft.Design", "CA1062:ValidateArgumentsOfPublicMethods")
public function MlsSettings (
    other : MlsSettings
)
```

Parameters

other

The existing instance.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
	 ReferenceEquals	(inherited from Object)
	ToString	Overridden. Creates a string representation of the current MlsSettings configuration element.

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ConfigurationElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[MlsSettings Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.ToString Method

Creates a string representation of the current MlsSettings configuration element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function ToString As String
```

C#

```
public override string ToString ()
```

C++

```
public:  
virtual String^ ToString () override
```

J#

```
public String ToString ()
```

JScript

```
public override function ToString () : String
```

Return Value

The current MlsSettings configuration element as a string.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Calendar	Gets or sets the calendar used in the management license settings.
	Comparison	Gets or sets the comparison used in the management license settings.
	Currency	Gets or sets the currency used in the management license settings.
	DateFormat	Gets or sets the date format used in the used in the management license settings.
	DateLanguage	Gets or sets the date format used in the management license settings.
	DualCurrency	Gets or sets the dual currency used in the management license settings.
	ElementInformation	(inherited from ConfigurationElement)
	ISOCurrency	Gets or sets the ISO currency used in the management license settings.
	Language	Gets or sets the language used in the management license settings.
	LengthSemantics	Gets or sets the length semantics used in the management license settings.
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	NCharConversionException	Gets or sets whether the specified object is a national character conversion exception.
	NumericCharacters	Gets or sets the numeric characters used in the management license settings.
	Sort	Gets or sets the sort order used in the management license settings.
	Territory	Gets or sets the territory used in the management license settings.
	TimeStampFormat	Gets or sets the time stamp format used in the management license settings.
	TimeStampTZFormat	Gets or sets the time stamp time zone format used in the management license settings.
	TimeZone	Gets or sets time zone used in the management license settings.

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	Overridden. Gets the properties used in the management license settings.

See Also

Reference

[MlsSettings Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Calendar Property

Gets or sets the calendar used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("calendar", DefaultValue:=Nothing)> _
Public Property Calendar As String
```

C#

```
[ConfigurationPropertyAttribute("calendar", DefaultValue=null)]
public string Calendar { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"calendar", DefaultValue=nullptr)]
public:
virtual property String^ Calendar {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Calendar ()

/** @property */
public final void set_Calendar (String value)
```

JScript

```
public final function get Calendar () : String
public final function set Calendar (value : String)
```

Property Value

The calendar used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Comparison Property

Gets or sets the comparison used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("comparison", DefaultValue:=Nothing)> _
Public Property Comparison As String
```

C#

```
[ConfigurationPropertyAttribute("comparison", DefaultValue=null)]
public string Comparison { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"comparison", DefaultValue=nullptr)]
public:
virtual property String^ Comparison {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Comparison ()

/** @property */
public final void set_Comparison (String value)
```

JScript

```
public final function get Comparison () : String

public final function set Comparison (value : String)
```

Property Value

The comparison used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Currency Property

Gets or sets the currency used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("currency", DefaultValue:=Nothing)> _
Public Property Currency As String
```

C#

```
[ConfigurationPropertyAttribute("currency", DefaultValue=null)]
public string Currency { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"currency", DefaultValue=nullptr)]
public:
virtual property String^ Currency {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Currency ()

/** @property */
public final void set_Currency (String value)
```

JScript

```
public final function get Currency () : String
public final function set Currency (value : String)
```

Property Value

The currency used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.DateFormat Property

Gets or sets the date format used in the used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("dateFormat", DefaultValue:=Nothing)> _
Public Property DateFormat As String
```

C#

```
[ConfigurationPropertyAttribute("dateFormat", DefaultValue=null)]
public string DateFormat { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"dateFormat", DefaultValue=nullptr)]
public:
virtual property String^ DateFormat {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_DateFormat ()

/** @property */
public final void set_DateFormat (String value)
```

JScript

```
public final function get DateFormat () : String

public final function set DateFormat (value : String)
```

Property Value

The date format used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.DateLanguage Property

Gets or sets the date format used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("dateLanguage", DefaultValue:=Nothing)> _
Public Property DateLanguage As String
```

C#

```
[ConfigurationPropertyAttribute("dateLanguage", DefaultValue=null)]
public string DateLanguage { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"dateLanguage", DefaultValue=nullptr)]
public:
virtual property String^ DateLanguage {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_DateLanguage ()

/** @property */
public final void set_DateLanguage (String value)
```

JScript

```
public final function get DateLanguage () : String

public final function set DateLanguage (value : String)
```

Property Value

The date format used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.DualCurrency Property

Gets or sets the dual currency used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("dualCurrency", DefaultValue:=Nothing)> _
Public Property DualCurrency As String
```

C#

```
[ConfigurationPropertyAttribute("dualCurrency", DefaultValue=null)]
public string DualCurrency { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"dualCurrency", DefaultValue=nullptr)]
public:
virtual property String^ DualCurrency {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_DualCurrency ()

/** @property */
public final void set_DualCurrency (String value)
```

JScript

```
public final function get DualCurrency () : String
public final function set DualCurrency (value : String)
```

Property Value

The dual currency used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.ISOCurrency Property

Gets or sets the ISO currency used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("iSOCurrency", DefaultValue:=Nothing)> _
Public Property ISOCurrency As String
```

C#

```
[ConfigurationPropertyAttribute("iSOCurrency", DefaultValue=null)]
public string ISOCurrency { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"iSOCurrency", DefaultValue=nullptr)]
public:
virtual property String^ ISOCurrency {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_ISOCurrency ()

/** @property */
public final void set_ISOCurrency (String value)
```

JScript

```
public final function get ISOCurrency () : String

public final function set ISOCurrency (value : String)
```

Property Value

The ISO currency used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Language Property

Gets or sets the language used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("language", DefaultValue:=Nothing)> _
Public Property Language As String
```

C#

```
[ConfigurationPropertyAttribute("language", DefaultValue=null)]
public string Language { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"language", DefaultValue=nullptr)]
public:
virtual property String^ Language {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Language ()

/** @property */
public final void set_Language (String value)
```

JScript

```
public final function get Language () : String

public final function set Language (value : String)
```

Property Value

The language used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.LengthSemantics Property

Gets or sets the length semantics used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("lengthSemantics", DefaultValue:=Nothing)> _
Public Property LengthSemantics As String
```

C#

```
[ConfigurationPropertyAttribute("lengthSemantics", DefaultValue=null)]
public string LengthSemantics { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"lengthSemantics", DefaultValue=nullptr)]
public:
virtual property String^ LengthSemantics {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_LengthSemantics ()

/** @property */
public final void set_LengthSemantics (String value)
```

JScript

```
public final function get LengthSemantics () : String
public final function set LengthSemantics (value : String)
```

Property Value

the length semantics used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.NCharConversionException Property

Gets or sets whether the specified object is a national character conversion exception.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("nCharConversionException", DefaultValue:=True)> _
Public Property NCharConversionException As Boolean
```

C#

```
[ConfigurationPropertyAttribute("nCharConversionException", DefaultValue=true)]
public bool NCharConversionException { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"nCharConversionException", DefaultValue=true)]
public:
virtual property bool NCharConversionException {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NCharConversionException ()

/** @property */
public final void set_NCharConversionException (boolean value)
```

JScript

```
public final function get NCharConversionException () : boolean
public final function set NCharConversionException (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object is a national conversion exception; otherwise, <languageKeyword>>false</languageKeyword>. The default value is <languageKeyword>>true</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.NumericCharacters Property

Gets or sets the numeric characters used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("numericCharacters", DefaultValue:=Nothing)> _
Public Property NumericCharacters As String
```

C#

```
[ConfigurationPropertyAttribute("numericCharacters", DefaultValue=null)]
public string NumericCharacters { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"numericCharacters", DefaultValue=nullptr)]
public:
virtual property String^ NumericCharacters {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NumericCharacters ()

/** @property */
public final void set_NumericCharacters (String value)
```

JScript

```
public final function get NumericCharacters () : String
public final function set NumericCharacters (value : String)
```

Property Value

The numeric characters used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Properties Property

Gets the properties used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Protected Overrides ReadOnly Property Properties As ConfigurationPropertyCollection
```

C#

```
protected override ConfigurationPropertyCollection Properties { get; }
```

C++

```
protected:
virtual property ConfigurationPropertyCollection^ Properties {
    ConfigurationPropertyCollection^ get () override;
}
```

J#

```
/** @property */
protected ConfigurationPropertyCollection get_Properties ()
```

JScript

```
protected override function get Properties () : ConfigurationPropertyCollection
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

MIsSettings Members
Microsoft.Adapters.OracleEBS Namespace

MlsSettings.Sort Property

Gets or sets the sort order used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("sort", DefaultValue:=Nothing)> _
Public Property Sort As String
```

C#

```
[ConfigurationPropertyAttribute("sort", DefaultValue=null)]
public string Sort { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"sort", DefaultValue=nullptr)]
public:
virtual property String^ Sort {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Sort ()

/** @property */
public final void set_Sort (String value)
```

JScript

```
public final function get Sort () : String

public final function set Sort (value : String)
```

Property Value

The sort used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.Territory Property

Gets or sets the territory used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("territory", DefaultValue:=Nothing)> _
Public Property Territory As String
```

C#

```
[ConfigurationPropertyAttribute("territory", DefaultValue=null)]
public string Territory { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"territory", DefaultValue=nullptr)]
public:
virtual property String^ Territory {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_Territory ()

/** @property */
public final void set_Territory (String value)
```

JScript

```
public final function get Territory () : String
public final function set Territory (value : String)
```

Property Value

The territory used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.TimeStampFormat Property

Gets or sets the time stamp format used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("timeStampFormat", DefaultValue:=Nothing)> _
Public Property TimeStampFormat As String
```

C#

```
[ConfigurationPropertyAttribute("timeStampFormat", DefaultValue=null)]
public string TimeStampFormat { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"timeStampFormat", DefaultValue=nullptr)]
public:
virtual property String^ TimeStampFormat {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_TimeStampFormat ()

/** @property */
public final void set_TimeStampFormat (String value)
```

JScript

```
public final function get TimeStampFormat () : String
public final function set TimeStampFormat (value : String)
```

Property Value

The time stamp format used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.TimeStampTZFormat Property

Gets or sets the time stamp time zone format used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("timeStampTZFormat", DefaultValue:=Nothing)> _
Public Property TimeStampTZFormat As String
```

C#

```
[ConfigurationPropertyAttribute("timeStampTZFormat", DefaultValue=null)]
public string TimeStampTZFormat { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"timeStampTZFormat", DefaultValue=nullptr)]
public:
virtual property String^ TimeStampTZFormat {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_TimeStampTZFormat ()

/** @property */
public final void set_TimeStampTZFormat (String value)
```

JScript

```
public final function get TimeStampTZFormat () : String
public final function set TimeStampTZFormat (value : String)
```

Property Value

The time stamp time zone format used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

MlsSettings.TimeZone Property

Gets or sets time zone used in the management license settings.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("timeZone", DefaultValue:=Nothing)> _
Public Property TimeZone As String
```

C#

```
[ConfigurationPropertyAttribute("timeZone", DefaultValue=null)]
public string TimeZone { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"timeZone", DefaultValue=nullptr)]
public:
virtual property String^ TimeZone {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_TimeZone ()

/** @property */
public final void set_TimeZone (String value)
```

JScript

```
public final function get TimeZone () : String

public final function set TimeZone (value : String)
```

Property Value

The time zone used in the management license settings.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[MlsSettings Class](#)

[MlsSettings Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter Class

The Binding Element for the Oracle E-Business Suite Adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public Class OracleEBSAdapter
    Inherits CommonAdapter
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public class OracleEBSAdapter : CommonAdapter
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSAdapter : public CommonAdapter
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public class OracleEBSAdapter extends CommonAdapter
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public class OracleEBSAdapter extends CommonAdapter
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.BindingElement](#)

[System.ServiceModel.Channels.TransportBindingElement](#)

[Microsoft.ServiceModel.Channels.Common.Adapter](#)

[Microsoft.Adapters.AdapterUtilities.CommonAdapter](#)

Microsoft.Adapters.OracleEBS.OracleEBSAdapter

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter Members

The Binding Element for the Oracle E-Business Suite Adapter.

The following tables list the members exposed by the [OracleEBSAdapter](#) type.

Public Constructors

Name	Description
OracleEBSAdapter	Initializes a new instance of the OracleApps class

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 ClientCredentialType	Gets or sets the client credential type.
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the specified object notifies on listener start.
 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.
 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.

 PolledDataAvailableStatement	Gets or sets the polled data available statement.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 Scheme	Overridden. Gets the scheme used in the adapter.
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated with a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 TransactedReceiveEnabled	(inherited from Adapter)
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Protected Properties

Name	Description
 Namespace	Overridden. Gets the namespace used for the adapter.
 Settings	(inherited from Adapter)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 BuildChannelFactory	(inherited from Adapter)
 BuildChannelListener	(inherited from Adapter)
 CanBuildChannelFactory	(inherited from Adapter)
 CanBuildChannelListener	(inherited from Adapter)
 Clone	(inherited from Adapter)
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetProperty	(inherited from TransportBindingElement)
 GetType	(inherited from Object)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 BuildConnectionFactory	Overridden. Initializes a channel factory for producing channels of a specified type from the specified object.
 BuildConnectionUri	Overridden. Builds the connection for the uniform resource identifier.
 BuildWsdRetrieval	Overridden. Builds an IWsdRetrieval object.
 CloneAdapter	Overridden. Returns a deep copy of the current Adapter object.
 Finalize	(inherited from Object)
 IsHandlerSupported	Overridden. Indicates whether the provided TConnectionHandler is supported by the current Adapter object.
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[OracleEBSAdapter Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter Constructor

Initializes a new instance of the OracleApps class

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSAdapter ()
```

C++

```
public:  
OracleEBSAdapter ()
```

J#

```
public OracleEBSAdapter ()
```

JScript

```
public function OracleEBSAdapter ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	BuildConnectionFactory	Overridden. Initializes a channel factory for producing channels of a specified type from the specified object.
	BuildConnectionUri	Overridden. Builds the connection for the uniform resource identifier.
	BuildWsdRetrieval	Overridden. Builds an IWsdRetrieval object.
	CloneAdapter	Overridden. Returns a deep copy of the current Adapter object.
	Finalize	(inherited from Object)
	IsHandlerSupported	Overridden. Indicates whether the provided TConnectionHandler is supported by the current Adapter object.
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleEBSAdapter Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.BuildConnectionFactory Method

Initializes a channel factory for producing channels of a specified type from the specified object.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Protected Overrides Function BuildConnectionFactory ( _
    connectionUri As ConnectionUri, _
    clientCredentials As ClientCredentials, _
    context As BindingContext _
) As IConnectionFactory
```

C#

```
protected override IConnectionFactory BuildConnectionFactory (
    ConnectionUri connectionUri,
    ClientCredentials clientCredentials,
    BindingContext context
)
```

C++

```
protected:
virtual IConnectionFactory^ BuildConnectionFactory (
    ConnectionUri^ connectionUri,
    ClientCredentials^ clientCredentials,
    BindingContext^ context
) override
```

J#

```
protected IConnectionFactory BuildConnectionFactory (
    ConnectionUri connectionUri,
    ClientCredentials clientCredentials,
    BindingContext context
)
```

JScript

```
protected override function BuildConnectionFactory (
    connectionUri : ConnectionUri,
    clientCredentials : ClientCredentials,
    context : BindingContext
) : IConnectionFactory
```

Parameters

connectionUri

The connection for the unified resource identifier.

clientCredentials

The client credentials

context

The specified context.

Return Value

The channel factory.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.BuildConnectionUri Method

Builds the connection for the uniform resource identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Protected Overrides Function BuildConnectionUri ( _
    uri As Uri _
) As ConnectionUri
```

C#

```
protected override ConnectionUri BuildConnectionUri (
    Uri uri
)
```

C++

```
protected:
virtual ConnectionUri^ BuildConnectionUri (
    Uri^ uri
) override
```

J#

```
protected ConnectionUri BuildConnectionUri (
    Uri uri
)
```

JScript

```
protected override function BuildConnectionUri (
    uri : Uri
) : ConnectionUri
```

Parameters

uri

The uniform resource identifier.

Return Value

The connection for the uniform resource identifier.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.BuildWsdRetrieval Method

Builds an IWsdRetrieval object.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Protected Overrides Function BuildWsdRetrieval ( _
    metadataLookup As MetadataLookup _
) As IWsdRetrieval
```

C#

```
protected override IWsdRetrieval BuildWsdRetrieval (
    MetadataLookup metadataLookup
)
```

C++

```
protected:
virtual IWsdRetrieval^ BuildWsdRetrieval (
    MetadataLookup^ metadataLookup
) override
```

J#

```
protected IWsdRetrieval BuildWsdRetrieval (
    MetadataLookup metadataLookup
)
```

JScript

```
protected override function BuildWsdRetrieval (
    metadataLookup : MetadataLookup
) : IWsdRetrieval
```

Parameters

metadataLookup

The specified MetadataLookup object.

Return Value

An IWsdRetrieval object using the specified MetadataLookup object.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.CloneAdapter Method

Returns a deep copy of the current Adapter object.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
Protected Overrides Function CloneAdapter As Adapter
```

C#

```
protected override Adapter CloneAdapter ()
```

C++

```
protected:
virtual Adapter^ CloneAdapter () override
```

J#

```
protected Adapter CloneAdapter ()
```

JScript

```
protected override function CloneAdapter () : Adapter
```

Return Value

An Adapter object containing a deep copy of the current Adapter.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.IsHandlerSupported Generic Method

Indicates whether the provided TConnectionHandler is supported by the current Adapter object.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Design", "CA1004:GenericMethodsShouldProvideTypeParameter")> _
Protected Overrides Function IsHandlerSupported(Of TConnectionHandler As IConnectionHandler
) As Boolean
```

C#

```
[SuppressMessageAttribute("Microsoft.Design", "CA1004:GenericMethodsShouldProvideTypeParameter")]
protected override bool IsHandlerSupported<TConnectionHandler> () where TConnectionHandler
: IConnectionHandler
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Design", L"CA1004:GenericMethodsShouldProvideTypeParameter")]
protected:
generic<typename TConnectionHandler> where TConnectionHandler : IConnectionHandler
virtual bool IsHandlerSupported () override
```

J#

JScript

Return Value

<languageKeyword>True</languageKeyword> if the provided TConnectionHandler is supported by the current Adapter object; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 ClientCredentialType	Gets or sets the client credential type.
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the specified object notifies on listener start.
 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.
 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.
 PolledDataAvailableStatement	Gets or sets the polled data available statement.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.

 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 Scheme	Overridden. Gets the scheme used in the adapter.
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated with a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 TransactedReceiveEnabled	(inherited from Adapter)
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNamespace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Protected Properties

	Name	Description
 Namespace		Overridden. Gets the namespace used for the adapter.
 Settings		(inherited from Adapter)

See Also

Reference

[OracleEBSAdapter Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.AcceptCredentialsInUri Property

Gets or sets whether the connection URI can contain user credentials for the database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property AcceptCredentialsInUri As Boolean
```

C#

```
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AcceptCredentialsInUri () : boolean

CompilerGeneratedAttribute
public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection URI can contain user credentials for the Oracle database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.ApplicationShortName Property

Gets or sets the application short name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property ApplicationShortName As String
```

C#

```
public string ApplicationShortName { get; set; }
```

C++

```
public:
virtual property String^ ApplicationShortName {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_ApplicationShortName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_ApplicationShortName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get ApplicationShortName () : String

CompilerGeneratedAttribute
public final function set ApplicationShortName (value : String)
```

Property Value

The application short name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.ClientCredentialType Property

Gets or sets the client credential type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property ClientCredentialType As ClientCredential
```

C#

```
public ClientCredential ClientCredentialType { get; set; }
```

C++

```
public:
virtual property ClientCredential ClientCredentialType {
    ClientCredential get () sealed;
    void set (ClientCredential value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final ClientCredential get_ClientCredentialType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_ClientCredentialType (ClientCredential value)
```

JScript

```
CompilerGeneratedAttribute
public final function get ClientCredentialType () : ClientCredential

CompilerGeneratedAttribute
public final function set ClientCredentialType (value : ClientCredential)
```

Property Value

The client credential type.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.ConnectionLifetime Property

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property ConnectionLifetime As Integer
```

C#

```
public int ConnectionLifetime { get; set; }
```

C++

```
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int

public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.DataFetchSize Property

Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property DataFetchSize As Long
```

C#

```
public long DataFetchSize { get; set; }
```

C++

```
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The amount of data in bytes that data provider fetches from the result set in one server round trip.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.DecrPoolSize Property

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property DecrPoolSize As Integer
```

C#

```
public int DecrPoolSize { get; set; }
```

C++

```
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.EnableBizTalkCompatibilityMode Property

Gets or sets whether the BizTalk compatibility mode is enabled.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
public:  
virtual property bool EnableBizTalkCompatibilityMode {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnableBizTalkCompatibilityMode ()  
  
/** @property */  
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean  
  
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the BizTalk compatibility mode is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.EnablePerformanceCounters Property

Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property EnablePerformanceCounters As Boolean
```

C#

```
public bool EnablePerformanceCounters { get; set; }
```

C++

```
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.EnableSafeTyping Property

Gets or sets whether enables or disables safe typing.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property EnableSafeTyping As Boolean
```

C#

```
public bool EnableSafeTyping { get; set; }
```

C++

```
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableSafeTyping ()

/** @property */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
public final function get EnableSafeTyping () : boolean

public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if safe typing is enabled; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.GeneratedUserTypesAssemblyFilePath Property

Gets or sets the name and path of the DLL that the adapter generates.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get GeneratedUserTypesAssemblyFilePath () : String

CompilerGeneratedAttribute
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.GeneratedUserTypesAssemblyKeyFilePath Property

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
public:  
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {  
    String^ get () sealed;  
    void set (String^ value) sealed;  
}
```

J#

```
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()  
  
/** @property */  
/** @attribute CompilerGeneratedAttribute() */  
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute  
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String  
  
CompilerGeneratedAttribute  
public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.InboundOperationType Property

Gets or sets the inbound operation type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property InboundOperationType As InboundOperation
```

C#

```
public InboundOperation InboundOperationType { get; set; }
```

C++

```
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The polling or Notification inbound operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.IncrPoolSize Property

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Property IncrPoolSize As Integer
```

C#

```
public int IncrPoolSize { get; set; }
```

C++

```
public:  
virtual property int IncrPoolSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_IncrPoolSize ()  
  
/** @property */  
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int  
  
public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.InsertBatchSize Property

Gets or sets the batch size for multiple record Insert operations.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property InsertBatchSize As Integer
```

C#

```
public int InsertBatchSize { get; set; }
```

C++

```
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int

public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for multiple record Insert operations.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.LongDatatypeColumnSize Property

NOTE: This property is now obsolete.

Gets or sets the maximum size in bytes of an Oracle long data type column.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")> _
Public Property LongDatatypeColumnSize As Integer
```

C#

```
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")]
public int LongDatatypeColumnSize { get; set; }
```

C++

```
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation
for suggested alternatives")]
public:
virtual property int LongDatatypeColumnSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (int value)
```

JScript

```
public final function get LongDatatypeColumnSize () : int
public final function set LongDatatypeColumnSize (value : int)
```

Property Value

The maximum size in bytes of an Oracle long data type column.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.MaxOutputAssociativeArrayElements Property

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
public:  
virtual property int MaxOutputAssociativeArrayElements {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_MaxOutputAssociativeArrayElements ()  
  
/** @property */  
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get MaxOutputAssociativeArrayElements () : int  
  
public final function set MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.MaxPoolSize Property

Gets or sets the maximum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property MaxPoolSize As Integer
```

C#

```
public int MaxPoolSize { get; set; }
```

C++

```
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.MetadataPooling Property

Gets or sets whether data provider caches metadata information for executed queries.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property MetadataPooling As Boolean
```

C#

```
public bool MetadataPooling { get; set; }
```

C++

```
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_MetadataPooling ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MetadataPooling (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MetadataPooling () : boolean

CompilerGeneratedAttribute
public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the data provider caches metadata information for executed queries; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.MinPoolSize Property

Gets or sets the minimum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property MinPoolSize As Integer
```

C#

```
public int MinPoolSize { get; set; }
```

C++

```
public:
virtual property int MinPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MinPoolSize ()

/** @property */
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int

public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.MlsSettings Property

Gets or sets the Multiple Language System Settings configuration element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property MlsSettings As MlsSettings
```

C#

```
public MlsSettings MlsSettings { get; set; }
```

C++

```
public:
virtual property MlsSettings^ MlsSettings {
    MlsSettings^ get () sealed;
    void set (MlsSettings^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final MlsSettings get_MlsSettings ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MlsSettings (MlsSettings value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MlsSettings () : MlsSettings

CompilerGeneratedAttribute
public final function set MlsSettings (value : MlsSettings)
```

Property Value

The Multiple Language System Settings configuration element.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.Namespace Property

Gets the namespace used for the adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Protected Overrides ReadOnly Property Namespace As String
```

C#

```
protected override string Namespace { get; }
```

C++

```
protected:
virtual property String^ Namespace {
    String^ get () override;
}
```

J#

```
/** @property */
protected String get_Namespace ()
```

JScript

```
protected override function get Namespace () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

OracleEBSAdapter Members
Microsoft.Adapters.OracleEBS Namespace

OracleEBSAdapter.NotificationPort Property

Gets or sets the port number that must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Property NotificationPort As Integer
```

C#

```
public int NotificationPort { get; set; }
```

C++

```
public:  
virtual property int NotificationPort {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_NotificationPort ()  
  
/** @property */  
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int  
  
public final function set NotificationPort (value : int)
```

Property Value

The port number that must open to listen for database change notification from Oracle database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.NotificationStatement Property

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property NotificationStatement As String
```

C#

```
public string NotificationStatement { get; set; }
```

C++

```
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

The SELECT statement used to register for getting notifications from Oracle database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.NotifyOnListenerStart Property

Gets or sets whether the specified object notifies on listener start.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property NotifyOnListenerStart As Boolean
```

C#

```
public bool NotifyOnListenerStart { get; set; }
```

C++

```
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> the specified object notifies on listener start; otherwise, <languageKeyword>False</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OracleConnectionClientId Property

Gets or sets the Oracle connection client identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OracleConnectionClientId As String
```

C#

```
public string OracleConnectionClientId { get; set; }
```

C++

```
public:
virtual property String^ OracleConnectionClientId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleConnectionClientId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleConnectionClientId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleConnectionClientId () : String

CompilerGeneratedAttribute
public final function set OracleConnectionClientId (value : String)
```

Property Value

The Oracle connection client identifier.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OracleEBSOrganizationId Property

Gets or sets the Oracle EBS organization identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OracleEBSOrganizationId As String
```

C#

```
public string OracleEBSOrganizationId { get; set; }
```

C++

```
public:
virtual property String^ OracleEBSOrganizationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSOrganizationId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSOrganizationId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSOrganizationId () : String

CompilerGeneratedAttribute
public final function set OracleEBSOrganizationId (value : String)
```

Property Value

The Oracle EBS organization identifier.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OracleEBSResponsibilityKey Property

Gets or sets the Oracle EBS responsibility key.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OracleEBSResponsibilityKey As String
```

C#

```
public string OracleEBSResponsibilityKey { get; set; }
```

C++

```
public:
virtual property String^ OracleEBSResponsibilityKey {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSResponsibilityKey ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSResponsibilityKey (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSResponsibilityKey () : String

CompilerGeneratedAttribute
public final function set OracleEBSResponsibilityKey (value : String)
```

Property Value

The Oracle EBS responsibility key.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OracleEBSResponsibilityName Property

Gets or sets the Oracle EBS responsibility name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OracleEBSResponsibilityName As String
```

C#

```
public string OracleEBSResponsibilityName { get; set; }
```

C++

```
public:
virtual property String^ OracleEBSResponsibilityName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSResponsibilityName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSResponsibilityName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSResponsibilityName () : String

CompilerGeneratedAttribute
public final function set OracleEBSResponsibilityName (value : String)
```

Property Value

The Oracle EBS responsibility name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OraclePassword Property

Gets or sets the Oracle password.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OraclePassword As String
```

C#

```
public string OraclePassword { get; set; }
```

C++

```
public:
virtual property String^ OraclePassword {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OraclePassword ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OraclePassword (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OraclePassword () : String

CompilerGeneratedAttribute
public final function set OraclePassword (value : String)
```

Property Value

The Oracle password.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.OracleUserName Property

Gets or sets the Oracle user name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property OracleUserName As String
```

C#

```
public string OracleUserName { get; set; }
```

C++

```
public:
virtual property String^ OracleUserName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleUserName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleUserName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleUserName () : String

CompilerGeneratedAttribute
public final function set OracleUserName (value : String)
```

Property Value

The Oracle user name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PolledDataAvailableStatement Property

Gets or sets the polled data available statement.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PolledDataAvailableStatement As String
```

C#

```
public string PolledDataAvailableStatement { get; set; }
```

C++

```
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The polled data available statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PollingAction Property

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PollingAction As String
```

C#

```
public string PollingAction { get; set; }
```

C++

```
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingAction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingAction (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingAction () : String

CompilerGeneratedAttribute
public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PollingInput Property

Gets or sets the polling input.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PollingInput As String
```

C#

```
public string PollingInput { get; set; }
```

C++

```
public:
virtual property String^ PollingInput {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingInput ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingInput (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingInput () : String

CompilerGeneratedAttribute
public final function set PollingInput (value : String)
```

Property Value

The polling input.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PollingInterval Property

Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PollingInterval As Integer
```

C#

```
public int PollingInterval { get; set; }
```

C++

```
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int

public final function set PollingInterval (value : int)
```

Property Value

The transacted polling interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PollWhileDataFound Property

Gets or sets whether the specified object continuously poll data when it is available.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PollWhileDataFound As Boolean
```

C#

```
public bool PollWhileDataFound { get; set; }
```

C++

```
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object continuously poll data when it is available; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.PostPollStatement Property

Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property PostPollStatement As String
```

C#

```
public string PostPollStatement { get; set; }
```

C++

```
public:
virtual property String^ PostPollStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PostPollStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PostPollStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PostPollStatement () : String

CompilerGeneratedAttribute
public final function set PostPollStatement (value : String)
```

Property Value

The PL/SQL block that is executed after the polling statement

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.Scheme Property

Gets the scheme used in the adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property Scheme As String
```

C#

```
public override string Scheme { get; }
```

C++

```
public:
virtual property String^ Scheme {
String^ get () override;
}
```

J#

```
/** @property */
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Property Value

The scheme used in the adapter.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.SkipNilNodes Property

Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property SkipNilNodes As Boolean
```

C#

```
public bool SkipNilNodes { get; set; }
```

C++

```
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_SkipNilNodes ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get SkipNilNodes () : boolean

CompilerGeneratedAttribute
public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the nil node is skipped; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.StatementCachePurge Property

Gets or sets whether the data provider statement cache associated with a connection is purged.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property StatementCachePurge As Boolean
```

C#

```
public bool StatementCachePurge { get; set; }
```

C++

```
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_StatementCachePurge ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get StatementCachePurge () : boolean

CompilerGeneratedAttribute
public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the statement cache purge is specified; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.StatementCacheSize Property

Gets or sets the maximum number of statements that can be cached by each data provider connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property StatementCacheSize As Integer
```

C#

```
public int StatementCacheSize { get; set; }
```

C++

```
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int

public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements that can be cached by each data provider connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.UseAmbientTransaction Property

Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property UseAmbientTransaction As Boolean
```

C#

```
public bool UseAmbientTransaction { get; set; }
```

C++

```
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter performs the operations using the transaction context provided by the caller; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.UseOracleConnectionPool Property

Gets or sets whether to use the Oracle connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property UseOracleConnectionPool As Boolean
```

C#

```
public bool UseOracleConnectionPool { get; set; }
```

C++

```
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseOracleConnectionPool ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseOracleConnectionPool () : boolean

CompilerGeneratedAttribute
public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection pool; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.UserAssembliesLoadPath Property

Gets or sets the path where the assemblies will be loaded.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property UserAssembliesLoadPath As String
```

C#

```
public string UserAssembliesLoadPath { get; set; }
```

C++

```
public:
virtual property String^ UserAssembliesLoadPath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_UserAssembliesLoadPath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UserAssembliesLoadPath () : String

CompilerGeneratedAttribute
public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The path where the assemblies will be loaded.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSAdapter.UseSchemaInNameSpace Property

Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Property UseSchemaInNameSpace As Boolean
```

C#

```
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseSchemaInNameSpace () : boolean

CompilerGeneratedAttribute
public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the schema name is included in the xml namespace for operations and their associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSAdapter Class](#)

[OracleEBSAdapter Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Class

Represents the Oracle E-Business Suite WCF Binding.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public Class OracleEBSBinding
    Inherits AdapterBinding
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public class OracleEBSBinding : AdapterBinding
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSBinding : public AdapterBinding
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public class OracleEBSBinding extends AdapterBinding
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public class OracleEBSBinding extends AdapterBinding
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.Binding](#)

[Microsoft.ServiceModel.Channels.Common.AdapterBinding](#)

Microsoft.Adapters.OracleEBS.OracleEBSBinding

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Members

Represents the Oracle E-Business Suite WCF Binding.

The following tables list the members exposed by the [OracleEBSBinding](#) type.

Public Constructors

Name	Description
OracleEBSBinding	Overloaded. Initializes a new instance of the OracleEBSBinding class.

Top

Public Properties

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 ClientCredentialType	Gets or sets the client credential type.
 CloseTimeout	(inherited from Binding)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 ConnectionUriType	Overridden. Gets the type of the specific ConnectionUri.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MessageVersion	(inherited from Binding)
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from Binding)
 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.

 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.
 PolledDataAvailableStatement	Gets or sets the polled data available statement.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Gets the scheme of the binding.
 SendTimeout	(inherited from Binding)
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated when a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 SupportsCertificateAuthentication	(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Gets a value that indicates whether the binding supports metadata browsing.
 SupportsMetadataGet	Overridden. Gets a value that indicates whether the binding supports metadata retrieval.
 SupportsMetadataSearch	Overridden. Gets a value that indicates whether the binding supports metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNamespace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates a clone of the existing BindingElement and returns it.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[OracleEBSBinding Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Constructor

Initializes a new instance of the [OracleEBSBinding](#) class.

Overload List

Name	Description
OracleEBSBinding ()	Initializes a new instance of the OracleEBSBinding class.
OracleEBSBinding (String)	Initializes a new instance of the OracleEBSBinding class with a configuration name.

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Constructor ()

Initializes a new instance of the [OracleEBSBinding](#) class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="
Microsoft.Adapters.OracleEBS.OracleEBSAdapter")> _
Public Sub New
```

C#

```
[SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="
Microsoft.Adapters.OracleEBS.OracleEBSAdapter")]
public OracleEBSBinding ()
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Usage", L"CA1806:DoNotIgnoreMethodResults", MessageId
=L"Microsoft.Adapters.OracleEBS.OracleEBSAdapter")]
public:
OracleEBSBinding ()
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults
", MessageId="Microsoft.Adapters.OracleEBS.OracleEBSAdapter") */
public OracleEBSBinding ()
```

JScript

```
SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="M
icrosoft.Adapters.OracleEBS.OracleEBSAdapter")
public function OracleEBSBinding ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Constructor (String)

Initializes a new instance of the [OracleEBSBinding](#) class with a configuration name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    configName As String _  
)
```

C#

```
public OracleEBSBinding (  
    string configName  
)
```

C++

```
public:  
OracleEBSBinding (  
    String^ configName  
)
```

J#

```
public OracleEBSBinding (  
    String configName  
)
```

JScript

```
public function OracleEBSBinding (  
    configName : String  
)
```

Parameters

configName

The name of the configuration.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates a clone of the existing BindingElement and returns it.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleEBSBinding Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.CreateBindingElements Method

Creates a clone of the existing BindingElement and returns it.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateBindingElements As BindingElementCollection
```

C#

```
public override BindingElementCollection CreateBindingElements ()
```

C++

```
public:  
virtual BindingElementCollection^ CreateBindingElements () override
```

J#

```
public BindingElementCollection CreateBindingElements ()
```

JScript

```
public override function CreateBindingElements () : BindingElementCollection
```

Return Value

The collection that contains the binding elements.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding Properties

Public Properties

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 ClientCredentialType	Gets or sets the client credential type.
 CloseTimeout	(inherited from Binding)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 ConnectionUriType	Overridden. Gets the type of the specific ConnectionUri.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MessageVersion	(inherited from Binding)
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from Binding)
 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.
 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.
 PolledDataAvailableStatement	Gets or sets the polled data available statement.

 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Gets the scheme of the binding.
 SendTimeout	(inherited from Binding)
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated when a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 SupportsCertificateAuthentication	(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Gets a value that indicates whether the binding supports metadata browsing.
 SupportsMetadataGet	Overridden. Gets a value that indicates whether the binding supports metadata retrieval.
 SupportsMetadataSearch	Overridden. Gets a value that indicates whether the binding supports metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

See Also

Reference

[OracleEBSBinding Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.AcceptCredentialsInUri Property

Gets or sets whether the connection URI can contain user credentials for the database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Security")> _
<BrowsableAttribute(True)> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[CategoryAttribute("Security")]
[BrowsableAttribute(true)]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[CategoryAttribute(L"Security")]
[BrowsableAttribute(true)]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AcceptCredentialsInUri () : boolean

CompilerGeneratedAttribute
public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection URI can contain user credentials for the Oracle database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.ApplicationShortName Property

Gets or sets the application short name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<BrowsableAttribute(True)> _
Public Property ApplicationShortName As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[BrowsableAttribute(true)]
public string ApplicationShortName { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[BrowsableAttribute(true)]
public:
virtual property String^ ApplicationShortName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_ApplicationShortName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_ApplicationShortName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get ApplicationShortName () : String

CompilerGeneratedAttribute
public final function set ApplicationShortName (value : String)
```

Property Value

The application short name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.ClientCredentialType Property

Gets or sets the client credential type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
Public Property ClientCredentialType As ClientCredential
```

C#

```
[BrowsableAttribute(true)]
public ClientCredential ClientCredentialType { get; set; }
```

C++

```
[BrowsableAttribute(true)]
public:
virtual property ClientCredential ClientCredentialType {
    ClientCredential get () sealed;
    void set (ClientCredential value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final ClientCredential get_ClientCredentialType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_ClientCredentialType (ClientCredential value)
```

JScript

```
CompilerGeneratedAttribute
public final function get ClientCredentialType () : ClientCredential

CompilerGeneratedAttribute
public final function set ClientCredentialType (value : ClientCredential)
```

Property Value

The client credential type.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.ConnectionLifetime Property

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property ConnectionLifetime As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int ConnectionLifetime { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int ConnectionLifetime {
int get () sealed;
void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int

public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.ConnectionUriType Property

Gets the type of the specific ConnectionUri.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property ConnectionUriType As Type
```

C#

```
public override Type ConnectionUriType { get; }
```

C++

```
public:
virtual property Type^ ConnectionUriType {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_ConnectionUriType ()
```

JScript

```
public override function get ConnectionUriType () : Type
```

Property Value

The type of the specific ConnectionUri.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.DataFetchSize Property

Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property DataFetchSize As Long
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public long DataFetchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The amount of data in bytes that data provider fetches from the result set in one server round trip.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.DecrPoolSize Property

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property DecrPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int DecrPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.EnableBizTalkCompatibilityMode Property

Gets or sets whether the BizTalk compatibility mode is enabled.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[BrowsableAttribute(true)]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[BrowsableAttribute(true)]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnableBizTalkCompatibilityMode () : boolean

CompilerGeneratedAttribute
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the BizTalk compatibility mode is enabled.; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.EnablePerformanceCounters Property

Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Diagnostics")> _
<BrowsableAttribute(True)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[BrowsableAttribute(true)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[BrowsableAttribute(true)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnablePerformanceCounters ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnablePerformanceCounters () : boolean

CompilerGeneratedAttribute
public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.EnableSafeTyping Property

Gets or sets whether enables or disables safe typing.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Metadata")> _
<BrowsableAttribute(True)> _
Public Property EnableSafeTyping As Boolean
```

C#

```
[CategoryAttribute("Metadata")]
[BrowsableAttribute(true)]
public bool EnableSafeTyping { get; set; }
```

C++

```
[CategoryAttribute(L"Metadata")]
[BrowsableAttribute(true)]
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnableSafeTyping ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnableSafeTyping () : boolean

CompilerGeneratedAttribute
public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if safe typing is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.GeneratedUserTypesAssemblyFilePath Property

Gets or sets the name and path of the DLL that the adapter generates.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(false, false)]
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(false, false)]
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get GeneratedUserTypesAssemblyFilePath () : String

CompilerGeneratedAttribute
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.GeneratedUserTypesAssemblyKeyFilePath Property

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(False, False)> _
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(false, false)]
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(false, false)]
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
public:
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String

CompilerGeneratedAttribute
public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.InboundOperationType Property

Gets or sets the inbound operation type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The Polling or Notification inbound operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.IncrPoolSize Property

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property IncrPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int IncrPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int IncrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_IncrPoolSize ()

/** @property */
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int

public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.InsertBatchSize Property

Gets or sets the batch size for multiple record Insert operations.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property InsertBatchSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public int InsertBatchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int

public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for multiple record Insert operations.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.LongDatatypeColumnSize Property

NOTE: This property is now obsolete.

Gets or sets the maximum size in bytes of an Oracle long data type column.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")> _
Public Property LongDatatypeColumnSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation f
or suggested alternatives")]
public int LongDatatypeColumnSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation
for suggested alternatives")]
public:
virtual property int LongDatatypeColumnSize {
int get () sealed;
void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (int value)
```

JScript

```
public final function get LongDatatypeColumnSize () : int
public final function set LongDatatypeColumnSize (value : int)
```

Property Value

The maximum size in bytes of an Oracle long data type column.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.MaxOutputAssociativeArrayElements Property

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("BufferManagement")> _
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("BufferManagement")]
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property int MaxOutputAssociativeArrayElements {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxOutputAssociativeArrayElements ()

/** @property */
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get_MaxOutputAssociativeArrayElements () : int
public final function set_MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.MaxPoolSize Property

Gets or sets the maximum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleConnectionPool")> _
<BrowsableAttribute(True)> _
Public Property MaxPoolSize As Integer
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[BrowsableAttribute(true)]
public int MaxPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[BrowsableAttribute(true)]
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.MetadataPooling Property

Gets or sets whether data provider caches metadata information for executed queries.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property MetadataPooling As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public bool MetadataPooling { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_MetadataPooling ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MetadataPooling (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MetadataPooling () : boolean

CompilerGeneratedAttribute
public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the data provider caches metadata information for executed queries; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.MinPoolSize Property

Gets or sets the minimum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MinPoolSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleConnectionPool")]
public int MinPoolSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MinPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MinPoolSize ()

/** @property */
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int

public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.MlsSettings Property

Gets or sets the Multiple Language System Settings configuration element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Multi Language Support")> _
<BrowsableAttribute(True)> _
Public Property MlsSettings As MlsSettings
```

C#

```
[CategoryAttribute("Multi Language Support")]
[BrowsableAttribute(true)]
public MlsSettings MlsSettings { get; set; }
```

C++

```
[CategoryAttribute(L"Multi Language Support")]
[BrowsableAttribute(true)]
public:
virtual property MlsSettings^ MlsSettings {
MlsSettings^ get () sealed;
void set (MlsSettings^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final MlsSettings get_MlsSettings ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MlsSettings (MlsSettings value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MlsSettings () : MlsSettings

CompilerGeneratedAttribute
public final function set MlsSettings (value : MlsSettings)
```

Property Value

The Multiple Language System Settings configuration element.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.NotificationPort Property

Gets or sets the port number that must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
<BrowsableAttribute(True)> _
Public Property NotificationPort As Integer
```

C#

```
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
public int NotificationPort { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
public:
virtual property int NotificationPort {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_NotificationPort ()

/** @property */
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int

public final function set NotificationPort (value : int)
```

Property Value

The port number that must open to listen for database change notification from Oracle database.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.NotificationStatement Property

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<BrowsableAttribute(True)> _
<CategoryAttribute("Notification")> _
Public Property NotificationStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
[CategoryAttribute("Notification")]
public string NotificationStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[BrowsableAttribute(true)]
[CategoryAttribute(L"Notification")]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

The SELECT statement used to register for getting notifications from Oracle database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.NotifyOnListenerStart Property

Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification")> _
<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[CategoryAttribute("Notification")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OracleConnectionClientId Property

Gets or sets the Oracle connection client identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Debug")> _
<BrowsableAttribute(True)> _
Public Property OracleConnectionClientId As String
```

C#

```
[CategoryAttribute("Debug")]
[BrowsableAttribute(true)]
public string OracleConnectionClientId { get; set; }
```

C++

```
[CategoryAttribute(L"Debug")]
[BrowsableAttribute(true)]
public:
virtual property String^ OracleConnectionClientId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleConnectionClientId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleConnectionClientId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleConnectionClientId () : String

CompilerGeneratedAttribute
public final function set OracleConnectionClientId (value : String)
```

Property Value

The Oracle connection client identifier.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OracleEBSOrganizationId Property

Gets or sets the Oracle EBS organization identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<BrowsableAttribute(True)> _
Public Property OracleEBSOrganizationId As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[BrowsableAttribute(true)]
public string OracleEBSOrganizationId { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[BrowsableAttribute(true)]
public:
virtual property String^ OracleEBSOrganizationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSOrganizationId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSOrganizationId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSOrganizationId () : String

CompilerGeneratedAttribute
public final function set OracleEBSOrganizationId (value : String)
```

Property Value

The Oracle EBS organization identifier.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OracleEBSResponsibilityKey Property

Gets or sets the Oracle EBS responsibility key.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<BrowsableAttribute(True)> _
Public Property OracleEBSResponsibilityKey As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[BrowsableAttribute(true)]
public string OracleEBSResponsibilityKey { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[BrowsableAttribute(true)]
public:
virtual property String^ OracleEBSResponsibilityKey {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSResponsibilityKey ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSResponsibilityKey (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSResponsibilityKey () : String

CompilerGeneratedAttribute
public final function set OracleEBSResponsibilityKey (value : String)
```

Property Value

The Oracle EBS responsibility key.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OracleEBSResponsibilityName Property

Gets or sets the Oracle EBS responsibility name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<BrowsableAttribute(True)> _
Public Property OracleEBSResponsibilityName As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[BrowsableAttribute(true)]
public string OracleEBSResponsibilityName { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[BrowsableAttribute(true)]
public:
virtual property String^ OracleEBSResponsibilityName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleEBSResponsibilityName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleEBSResponsibilityName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleEBSResponsibilityName () : String

CompilerGeneratedAttribute
public final function set OracleEBSResponsibilityName (value : String)
```

Property Value

The Oracle EBS responsibility name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OraclePassword Property

Gets or sets the Oracle password.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<BrowsableAttribute(True)> _
<PasswordPropertyTextAttribute(True)> _
Public Property OraclePassword As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[BrowsableAttribute(true)]
[PasswordPropertyTextAttribute(true)]
public string OraclePassword { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[BrowsableAttribute(true)]
[PasswordPropertyTextAttribute(true)]
public:
virtual property String^ OraclePassword {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OraclePassword ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OraclePassword (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OraclePassword () : String

CompilerGeneratedAttribute
public final function set OraclePassword (value : String)
```

Property Value

The Oracle password.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.OracleUserName Property

Gets or sets the Oracle user name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("OracleEBS")> _
Public Property OracleUserName As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("OracleEBS")]
public string OracleUserName { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"OracleEBS")]
public:
virtual property String^ OracleUserName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_OracleUserName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_OracleUserName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get OracleUserName () : String

CompilerGeneratedAttribute
public final function set OracleUserName (value : String)
```

Property Value

The Oracle user name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PolledDataAvailableStatement Property

Gets or sets the polled data available statement.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The polled data available statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PollingAction Property

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
<BrowsableAttribute(True)> _
Public Property PollingAction As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
[BrowsableAttribute(true)]
public string PollingAction { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
[BrowsableAttribute(true)]
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingAction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingAction (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingAction () : String

CompilerGeneratedAttribute
public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PollingInput Property

Gets or sets the polling input.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Polling")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PollingInput As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Polling")]
[BindingPropertyDirectionAttribute(true, false)]
public string PollingInput { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Polling")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PollingInput {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingInput ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingInput (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingInput () : String

CompilerGeneratedAttribute
public final function set PollingInput (value : String)
```

Property Value

The polling input.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PollingInterval Property

Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
Public Property PollingInterval As Integer
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
public int PollingInterval { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int

public final function set PollingInterval (value : int)
```

Property Value

The transacted polling interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PollWhileDataFound Property

Gets or sets whether the specified object continuously poll data when it is available.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object continuously poll data when it is available; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.PostPollStatement Property

Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
Public Property PostPollStatement As String
```

C#

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
public string PostPollStatement { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
public:
virtual property String^ PostPollStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PostPollStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PostPollStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PostPollStatement () : String

CompilerGeneratedAttribute
public final function set PostPollStatement (value : String)
```

Property Value

The PL/SQL block that is executed after the polling statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.Scheme Property

Gets the scheme of the binding.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property Scheme As String
```

C#

```
public override string Scheme { get; }
```

C++

```
public:
virtual property String^ Scheme {
String^ get () override;
}
```

J#

```
/** @property */
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Property Value

The scheme of the binding.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.SkipNilNodes Property

Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Run Time Behavior")> _
<BrowsableAttribute(True)> _
Public Property SkipNilNodes As Boolean
```

C#

```
[CategoryAttribute("Run Time Behavior")]
[BrowsableAttribute(true)]
public bool SkipNilNodes { get; set; }
```

C++

```
[CategoryAttribute(L"Run Time Behavior")]
[BrowsableAttribute(true)]
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_SkipNilNodes ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get SkipNilNodes () : boolean

CompilerGeneratedAttribute
public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the nil node is skipped; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.StatementCachePurge Property

Gets or sets whether the data provider statement cache associated when a connection is purged.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<BrowsableAttribute(True)> _
Public Property StatementCachePurge As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[BrowsableAttribute(true)]
public bool StatementCachePurge { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[BrowsableAttribute(true)]
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_StatementCachePurge ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get StatementCachePurge () : boolean

CompilerGeneratedAttribute
public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the data provider statement cache associated with a connection is purged; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.StatementCacheSize Property

Gets or sets the maximum number of statements that can be cached by each data provider connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("BufferManagement")> _
Public Property StatementCacheSize As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("BufferManagement")]
public int StatementCacheSize { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int

public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements that can be cached by each data provider connection.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.SupportsMetadataBrowse Property

Gets a value that indicates whether the binding supports metadata browsing.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataBrowse As Boolean
```

C#

```
public override bool SupportsMetadataBrowse { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataBrowse {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataBrowse ()
```

JScript

```
public override function get SupportsMetadataBrowse () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the binding supports metadata browsing; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.SupportsMetadataGet Property

Gets a value that indicates whether the binding supports metadata retrieval.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataGet As Boolean
```

C#

```
public override bool SupportsMetadataGet { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataGet {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataGet ()
```

JScript

```
public override function get SupportsMetadataGet () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the binding supports metadata retrieval; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.SupportsMetadataSearch Property

Gets a value that indicates whether the binding supports metadata search.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataSearch As Boolean
```

C#

```
public override bool SupportsMetadataSearch { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataSearch {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataSearch ()
```

JScript

```
public override function get SupportsMetadataSearch () : boolean
```

Property Value

<languageKeyword>True</languageKeyword> if the binding supports metadata search; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.UseAmbientTransaction Property

Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Transactions")> _
<BrowsableAttribute(True)> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[CategoryAttribute("Transactions")]
[BrowsableAttribute(true)]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[CategoryAttribute(L"Transactions")]
[BrowsableAttribute(true)]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseAmbientTransaction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseAmbientTransaction () : boolean

CompilerGeneratedAttribute
public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter performs the operations using the transaction context provided by the caller; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.UseOracleConnectionPool Property

Gets or sets whether to use the Oracle connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleConnectionPool")> _
<BrowsableAttribute(True)> _
Public Property UseOracleConnectionPool As Boolean
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[BrowsableAttribute(true)]
public bool UseOracleConnectionPool { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[BrowsableAttribute(true)]
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseOracleConnectionPool ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseOracleConnectionPool () : boolean

CompilerGeneratedAttribute
public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection pool; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.UserAssembliesLoadPath Property

Gets or sets the path where the assemblies will be loaded.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("UDT .NET Type Generation - Run Time")> _
Public Property UserAssembliesLoadPath As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("UDT .NET Type Generation - Run Time")]
public string UserAssembliesLoadPath { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute("UDT .NET Type Generation - Run Time")]
public:
virtual property String^ UserAssembliesLoadPath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_UserAssembliesLoadPath ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UserAssembliesLoadPath () : String

CompilerGeneratedAttribute
public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The path where the assemblies will be loaded.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBinding.UseSchemaInNameSpace Property

Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Metadadata")> _
<BrowsableAttribute(True)> _
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
[CategoryAttribute("Metadadata")]
[BrowsableAttribute(true)]
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
[CategoryAttribute(L"Metadadata")]
[BrowsableAttribute(true)]
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseSchemaInNameSpace () : boolean

CompilerGeneratedAttribute
public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the schema name is included in the xml namespace for operations and their associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBinding Class](#)

[OracleEBSBinding Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingCollectionElement Class

Represents the StandardBindingCollectionElement for the Oracle E-Business Suite Adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public Class OracleEBSBindingCollectionElement
    Inherits StandardBindingCollectionElement(Of OracleEBSBinding, OracleEBSBindingConfigurati
onElement)
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public class OracleEBSBindingCollectionElement : StandardBindingCollectionElement<OracleEBS
Binding,OracleEBSBindingConfigurationElement>
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSBindingCollectionElement : public StandardBindingCollectionElemen
t<OracleEBSBinding^, OracleEBSBindingConfigurationElement^>
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePas
calCased") */
public class OracleEBSBindingCollectionElement extends StandardBindingCollectionElement<Ora
cleEBSBinding,OracleEBSBindingConfigurationElement>
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public class OracleEBSBindingCollectionElement extends StandardBindingCollectionElement<Ora
cleEBSBinding,OracleEBSBindingConfigurationElement>
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.BindingCollectionElement](#)

[System.ServiceModel.Configuration.StandardBindingCollectionElement](#)

Microsoft.Adapters.OracleEBS.OracleEBSBindingCollectionElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingCollectionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingCollectionElement Members

Represents the StandardBindingCollectionElement for the Oracle E-Business Suite Adapter.

The following tables list the members exposed by the [OracleEBSBindingCollectionElement](#) type.

Public Constructors

Name	Description
OracleEBSBindingCollectionElement	Initializes a new instance of the OracleEBSBindingCollectionElement class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BindingName	(inherited from BindingCollectionElement)
 Bindings	(inherited from StandardBindingCollectionElement)
 BindingType	(inherited from StandardBindingCollectionElement)
 ConfiguredBindings	(inherited from StandardBindingCollectionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from StandardBindingCollectionElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 ContainsKey	(inherited from StandardBindingCollectionElement)
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 GetDefault	(inherited from StandardBindingCollectionElement)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)

 Reset	(inherited from ConfigurationElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 TryAdd	(inherited from StandardBindingCollectionElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[OracleEBSBindingCollectionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingCollectionElement Constructor

Initializes a new instance of the [OracleEBSBindingCollectionElement](#) class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSBindingCollectionElement ()
```

C++

```
public:  
OracleEBSBindingCollectionElement ()
```

J#

```
public OracleEBSBindingCollectionElement ()
```

JScript

```
public function OracleEBSBindingCollectionElement ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingCollectionElement Class](#)

[OracleEBSBindingCollectionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingCollectionElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ContainsKey	(inherited from StandardBindingCollectionElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	GetDefault	(inherited from StandardBindingCollectionElement)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ConfigurationElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	TryAdd	(inherited from StandardBindingCollectionElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[OracleEBSBindingCollectionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingCollectionElement Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	BindingName	(inherited from BindingCollectionElement)
	Bindings	(inherited from StandardBindingCollectionElement)
	BindingType	(inherited from StandardBindingCollectionElement)
	ConfiguredBindings	(inherited from StandardBindingCollectionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingCollectionElement)

See Also

Reference

[OracleEBSBindingCollectionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Class

Represents the StandardBindingElement for the Oracle E-Business Suite Adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public NotInheritable Class OracleEBSBindingConfigurationElement
    Inherits StandardBindingElement
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public sealed class OracleEBSBindingConfigurationElement : StandardBindingElement
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSBindingConfigurationElement sealed : public StandardBindingElement
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public final class OracleEBSBindingConfigurationElement extends StandardBindingElement
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public final class OracleEBSBindingConfigurationElement extends StandardBindingElement
```

Remarks

This class is used for configuring the Oracle E-Business Suite Adapter via .config files.

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.StandardBindingElement](#)

Microsoft.Adapters.OracleEBS.OracleEBSBindingConfigurationElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Members

Represents the StandardBindingElement for the Oracle E-Business Suite Adapter.

The following tables list the members exposed by the [OracleEBSBindingConfigurationElement](#) type.

Public Constructors

Name	Description
OracleEBSBindingConfigurationElement	Overloaded.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 ClientCredentialType	Gets or sets the client credential type.
 CloseTimeout	(inherited from StandardBindingElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 Name	(inherited from StandardBindingElement)
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from StandardBindingElement)

 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.
 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.
 PolledDataAvailableStatement	Gets or sets the polled data available statement.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated when a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

[Top](#)

Protected Properties

Name	Description
 BindingElementType	(inherited from StandardBindingElement)
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from StandardBindingElement)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 ApplyConfiguration	(inherited from StandardBindingElement)
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)

 InitializeFrom	(inherited from StandardBindingElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnApplyConfiguration	(inherited from StandardBindingElement)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from StandardBindingElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Constructor

Overload List

Name	Description
OracleEBSBindingConfigurationElement ()	Initializes a new instance of the OracleEBSBindingConfigurationElement class.
OracleEBSBindingConfigurationElement (String)	Initializes a new instance of the OracleEBSBindingConfigurationElement class with the specified name.

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Constructor ()

Initializes a new instance of the [OracleEBSBindingConfigurationElement](#) class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSBindingConfigurationElement ()
```

C++

```
public:  
OracleEBSBindingConfigurationElement ()
```

J#

```
public OracleEBSBindingConfigurationElement ()
```

JScript

```
public function OracleEBSBindingConfigurationElement ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Constructor (String)

Initializes a new instance of the [OracleEBSBindingConfigurationElement](#) class with the specified name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Sub New ( _
    name As String _
)
```

C#

```
public OracleEBSBindingConfigurationElement (
    string name
)
```

C++

```
public:
    OracleEBSBindingConfigurationElement (
        String^ name
    )
```

J#

```
public OracleEBSBindingConfigurationElement (
    String name
)
```

JScript

```
public function OracleEBSBindingConfigurationElement (
    name : String
)
```

Parameters

name

The specified name.

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from StandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnApplyConfiguration	(inherited from StandardBindingElement)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from StandardBindingElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets whether the connection URI can contain user credentials for the data base.
 ApplicationShortName	Gets or sets the application short name.
 ClientCredentialType	Gets or sets the client credential type.
 CloseTimeout	(inherited from StandardBindingElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets whether the BizTalk compatibility mode is enabled.
 EnablePerformanceCounters	Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets whether enables or disables safe typing.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the inbound operation type.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in the connection pool.
 MetadataPooling	Gets or sets whether data provider caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in the connection pool.
 MlsSettings	Gets or sets the Multiple Language System Settings configuration element.
 Name	(inherited from StandardBindingElement)
 NotificationPort	Gets or sets the port number that must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OpenTimeout	(inherited from StandardBindingElement)
 OracleConnectionClientId	Gets or sets the Oracle connection client identifier.
 OracleEBSOrganizationId	Gets or sets the Oracle EBS organization identifier.
 OracleEBSResponsibilityKey	Gets or sets the Oracle EBS responsibility key.
 OracleEBSResponsibilityName	Gets or sets the Oracle EBS responsibility name.
 OraclePassword	Gets or sets the Oracle password.
 OracleUserName	Gets or sets the Oracle user name.
 PolledDataAvailableStatement	Gets or sets the polled data available statement.

 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling input.
 PollingInterval	Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets whether the specified object continuously poll data when it is available.
 PostPollStatement	Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 SkipNilNodes	Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets whether the data provider statement cache associated when a connection is purged.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each data provider connection.
 UseAmbientTransaction	Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets whether to use the Oracle connection pool.
 UserAssembliesLoadPath	Gets or sets the path where the assemblies will be loaded.
 UseSchemaInNameSpace	Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Protected Properties

	Name	Description
	BindingElementType	(inherited from StandardBindingElement)
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingElement)

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)
[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.AcceptCredentialsInUri Property

Gets or sets whether the connection URI can contain user credentials for the database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Security")> _
<BrowsableAttribute(False)> _
<ConfigurationPropertyAttribute("acceptCredentialsInUri")> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[CategoryAttribute("Security")]
[BrowsableAttribute(false)]
[ConfigurationPropertyAttribute("acceptCredentialsInUri")]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[CategoryAttribute(L"Security")]
[BrowsableAttribute(false)]
[ConfigurationPropertyAttribute(L"acceptCredentialsInUri")]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection URI can contain user credentials for the Oracle database; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.ApplicationShortName Property

Gets or sets the application short name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleEBS")> _
<ConfigurationPropertyAttribute("applicationShortName")> _
Public Property ApplicationShortName As String
```

C#

```
[CategoryAttribute("OracleEBS")]
[ConfigurationPropertyAttribute("applicationShortName")]
public string ApplicationShortName { get; set; }
```

C++

```
[CategoryAttribute(L"OracleEBS")]
[ConfigurationPropertyAttribute(L"applicationShortName")]
public:
virtual property String^ ApplicationShortName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_ApplicationShortName ()

/** @property */
public final void set_ApplicationShortName (String value)
```

JScript

```
public final function get ApplicationShortName () : String
public final function set ApplicationShortName (value : String)
```

Property Value

The application short name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.ClientCredentialType Property

Gets or sets the client credential type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("clientCredentialType")> _
Public Property ClientCredentialType As ClientCredential
```

C#

```
[ConfigurationPropertyAttribute("clientCredentialType")]
public ClientCredential ClientCredentialType { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"clientCredentialType")]
public:
virtual property ClientCredential ClientCredentialType {
    ClientCredential get () sealed;
    void set (ClientCredential value) sealed;
}
```

J#

```
/** @property */
public final ClientCredential get_ClientCredentialType ()

/** @property */
public final void set_ClientCredentialType (ClientCredential value)
```

JScript

```
public final function get ClientCredentialType () : ClientCredential

public final function set ClientCredentialType (value : ClientCredential)
```

Property Value

The client credential type.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.ConnectionLifetime Property

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("connectionLifetime")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property ConnectionLifetime As Integer
```

C#

```
[ConfigurationPropertyAttribute("connectionLifetime")]
[CategoryAttribute("OracleConnectionPool")]
public int ConnectionLifetime { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"connectionLifetime")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int
public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.DataFetchSize Property

Gets or sets the amount of data in bytes that data provider fetches from the result set in one server round trip.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("dataFetchSize")> _
Public Property DataFetchSize As Long
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("dataFetchSize")]
public long DataFetchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"dataFetchSize")]
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The amount of data in bytes that data provider fetches from the result set in one server round trip.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.DecrPoolSize Property

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("OracleConnectionPool")> _
<ConfigurationPropertyAttribute("decrPoolSize")> _
Public Property DecrPoolSize As Integer
```

C#

```
[CategoryAttribute("OracleConnectionPool")]
[ConfigurationPropertyAttribute("decrPoolSize")]
public int DecrPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"OracleConnectionPool")]
[ConfigurationPropertyAttribute(L"decrPoolSize")]
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.EnableBizTalkCompatibilityMode Property

Gets or sets whether the BizTalk compatibility mode is enabled.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode")> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode")]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableBizTalkCompatibilityMode")]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean

public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the BizTalk compatibility mode is enabled; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)
[OracleEBSBindingConfigurationElement Members](#)
[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.EnablePerformanceCounters Property

Gets or sets whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Diagnostics")> _
<ConfigurationPropertyAttribute("enablePerformanceCounters")> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[ConfigurationPropertyAttribute("enablePerformanceCounters")]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[ConfigurationPropertyAttribute(L"enablePerformanceCounters")]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional

with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.EnableSafeTyping Property

Gets or sets whether enables or disables safe typing.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enableSafeTyping")> _
<CategoryAttribute("Metadata")> _
Public Property EnableSafeTyping As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableSafeTyping")]
[CategoryAttribute("Metadata")]
public bool EnableSafeTyping { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableSafeTyping")]
[CategoryAttribute(L"Metadata")]
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableSafeTyping ()

/** @property */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
public final function get EnableSafeTyping () : boolean

public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if safe typing is enabled; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.GeneratedUserTypesAssemblyFilePath Property

Gets or sets the name and path of the DLL that the adapter generates.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath")> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath")]
[BindingPropertyDirectionAttribute(false, false)]
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyFilePath")]
[BindingPropertyDirectionAttribute(false, false)]
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyFilePath () : String

public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

OracleEBSBindingConfigurationElement.GeneratedUserTypesAssemblyKeyFilePath Property

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath")> _
<CategoryAttribute("UDT .NET Type Generation - Design Time")> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath")]
[CategoryAttribute("UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyKeyFilePath")]
[CategoryAttribute(L"UDT .NET Type Generation - Design Time")]
[BindingPropertyDirectionAttribute(false, false)]
public:
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String

public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.InboundOperationType Property

Gets or sets the inbound operation type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("inboundOperationType")> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("inboundOperationType")]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"inboundOperationType")]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
public final InboundOperation get_InboundOperationType ()

/** @property */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
public final function get InboundOperationType () : InboundOperation
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

The Polling or Notification inbound operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.IncrPoolSize Property

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("incrPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property IncrPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("incrPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int IncrPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"incrPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int IncrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_IncrPoolSize ()

/** @property */
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int
public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.InsertBatchSize Property

Gets or sets the batch size for multiple record Insert operations.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("insertBatchSize")> _
Public Property InsertBatchSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("insertBatchSize")]
public int InsertBatchSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"insertBatchSize")]
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int
public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for multiple record Insert operations.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.LongDatatypeColumnSize Property

NOTE: This property is now obsolete.

Gets or sets the maximum size in bytes of an Oracle long data type column.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("longDatatypeColumnSize")> _
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")> _
Public Property LongDatatypeColumnSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("longDatatypeColumnSize")]
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")]
public int LongDatatypeColumnSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"longDatatypeColumnSize")]
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")]
public:
virtual property int LongDatatypeColumnSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (int value)
```

JScript

```
public final function get LongDatatypeColumnSize () : int
public final function set LongDatatypeColumnSize (value : int)
```

Property Value

The maximum size in bytes of an Oracle long data type column.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.MaxOutputAssociativeArrayElements Property

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements")> _
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements")]
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"maxOutputAssociativeArrayElements")]
public:
virtual property int MaxOutputAssociativeArrayElements {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxOutputAssociativeArrayElements ()

/** @property */
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get MaxOutputAssociativeArrayElements () : int
public final function set MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.MaxPoolSize Property

Gets or sets the maximum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("maxPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MaxPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int MaxPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.MetadataPooling Property

Gets or sets whether data provider caches metadata information for executed queries.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("metadataPooling")> _
<CategoryAttribute("BufferManagement")> _
Public Property MetadataPooling As Boolean
```

C#

```
[ConfigurationPropertyAttribute("metadataPooling")]
[CategoryAttribute("BufferManagement")]
public bool MetadataPooling { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"metadataPooling")]
[CategoryAttribute(L"BufferManagement")]
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_MetadataPooling ()

/** @property */
public final void set_MetadataPooling (boolean value)
```

JScript

```
public final function get MetadataPooling () : boolean

public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the data provider caches metadata information for executed queries; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.MinPoolSize Property

Gets or sets the minimum number of connections in the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("minPoolSize")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property MinPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("minPoolSize")]
[CategoryAttribute("OracleConnectionPool")]
public int MinPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"minPoolSize")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property int MinPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MinPoolSize ()

/** @property */
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int
public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections in the connection pool.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.MlsSettings Property

Gets or sets the Multiple Language System Settings configuration element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Multi Language Support")> _
<ConfigurationPropertyAttribute("mIsSettings")> _
Public Property MIsSettings As MIsSettings
```

C#

```
[CategoryAttribute("Multi Language Support")]
[ConfigurationPropertyAttribute("mIsSettings")]
public MIsSettings MIsSettings { get; set; }
```

C++

```
[CategoryAttribute(L"Multi Language Support")]
[ConfigurationPropertyAttribute(L"mIsSettings")]
public:
virtual property MIsSettings^ MIsSettings {
    MIsSettings^ get () sealed;
    void set (MIsSettings^ value) sealed;
}
```

J#

```
/** @property */
public final MIsSettings get_MIsSettings ()

/** @property */
public final void set_MIsSettings (MIsSettings value)
```

JScript

```
public final function get MIsSettings () : MIsSettings

public final function set MIsSettings (value : MIsSettings)
```

Property Value

The Multiple Language System Settings configuration element.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.NotificationPort Property

Gets or sets the port number that must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("notificationPort")> _
<CategoryAttribute("Notification")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotificationPort As Integer
```

C#

```
[ConfigurationPropertyAttribute("notificationPort")]
[CategoryAttribute("Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public int NotificationPort { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notificationPort")]
[CategoryAttribute(L"Notification")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property int NotificationPort {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_NotificationPort ()

/** @property */
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int

public final function set NotificationPort (value : int)
```

Property Value

The port number that must open to listen for database change notification from Oracle database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.NotificationStatement Property

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification")> _
<ConfigurationPropertyAttribute("notificationStatement")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotificationStatement As String
```

C#

```
[CategoryAttribute("Notification")]
[ConfigurationPropertyAttribute("notificationStatement")]
[BindingPropertyDirectionAttribute(true, false)]
public string NotificationStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[ConfigurationPropertyAttribute(L"notificationStatement")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NotificationStatement ()

/** @property */
public final void set_NotificationStatement (String value)
```

JScript

```
public final function get NotificationStatement () : String
public final function set NotificationStatement (value : String)
```

Property Value

The SELECT statement used to register for getting notifications from Oracle database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.NotifyOnListenerStart Property

Gets or sets whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification")> _
<ConfigurationPropertyAttribute("notifyOnListenerStart")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[CategoryAttribute("Notification")]
[ConfigurationPropertyAttribute("notifyOnListenerStart")]
[BindingPropertyDirectionAttribute(true, false)]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[CategoryAttribute(L"Notification")]
[ConfigurationPropertyAttribute(L"notifyOnListenerStart")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NotifyOnListenerStart ()

/** @property */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
public final function get NotifyOnListenerStart () : boolean

public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the adapter sends a notification message to the adapter clients, informing

that the receive location is running, when the listener starts; otherwise, <languageKeyword>>false</languageKeyword>. Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OracleConnectionClientId Property

Gets or sets the Oracle connection client identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("oracleConnectionClientId")> _
<CategoryAttribute("Debug")> _
Public Property OracleConnectionClientId As String
```

C#

```
[ConfigurationPropertyAttribute("oracleConnectionClientId")]
[CategoryAttribute("Debug")]
public string OracleConnectionClientId { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleConnectionClientId")]
[CategoryAttribute(L"Debug")]
public:
virtual property String^ OracleConnectionClientId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleConnectionClientId ()

/** @property */
public final void set_OracleConnectionClientId (String value)
```

JScript

```
public final function get OracleConnectionClientId () : String

public final function set OracleConnectionClientId (value : String)
```

Property Value

The Oracle connection client identifier.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OracleEBSOrganizationId Property

Gets or sets the Oracle EBS organization identifier.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("oracleEBSOrganizationId")> _
<CategoryAttribute("OracleEBS")> _
Public Property OracleEBSOrganizationId As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSOrganizationId")]
[CategoryAttribute("OracleEBS")]
public string OracleEBSOrganizationId { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSOrganizationId")]
[CategoryAttribute(L"OracleEBS")]
public:
virtual property String^ OracleEBSOrganizationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSOrganizationId ()

/** @property */
public final void set_OracleEBSOrganizationId (String value)
```

JScript

```
public final function get OracleEBSOrganizationId () : String
public final function set OracleEBSOrganizationId (value : String)
```

Property Value

The Oracle EBS organization identifier.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OracleEBSResponsibilityKey Property

Gets or sets the Oracle EBS responsibility key.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("oracleEBSResponsibilityKey")> _
<CategoryAttribute("OracleEBS")> _
Public Property OracleEBSResponsibilityKey As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSResponsibilityKey")]
[CategoryAttribute("OracleEBS")]
public string OracleEBSResponsibilityKey { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSResponsibilityKey")]
[CategoryAttribute(L"OracleEBS")]
public:
virtual property String^ OracleEBSResponsibilityKey {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSResponsibilityKey ()

/** @property */
public final void set_OracleEBSResponsibilityKey (String value)
```

JScript

```
public final function get OracleEBSResponsibilityKey () : String
public final function set OracleEBSResponsibilityKey (value : String)
```

Property Value

The Oracle EBS responsibility key.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OracleEBSResponsibilityName Property

Gets or sets the Oracle EBS responsibility name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("oracleEBSResponsibilityName")> _
<CategoryAttribute("OracleEBS")> _
Public Property OracleEBSResponsibilityName As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSResponsibilityName")]
[CategoryAttribute("OracleEBS")]
public string OracleEBSResponsibilityName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSResponsibilityName")]
[CategoryAttribute(L"OracleEBS")]
public:
virtual property String^ OracleEBSResponsibilityName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSResponsibilityName ()

/** @property */
public final void set_OracleEBSResponsibilityName (String value)
```

JScript

```
public final function get OracleEBSResponsibilityName () : String
public final function set OracleEBSResponsibilityName (value : String)
```

Property Value

The Oracle EBS responsibility name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003

with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OraclePassword Property

Gets or sets the Oracle password.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<PasswordPropertyTextAttribute(True)> _
<CategoryAttribute("OracleEBS")> _
<ConfigurationPropertyAttribute("oraclePassword")> _
Public Property OraclePassword As String
```

C#

```
[PasswordPropertyTextAttribute(true)]
[CategoryAttribute("OracleEBS")]
[ConfigurationPropertyAttribute("oraclePassword")]
public string OraclePassword { get; set; }
```

C++

```
[PasswordPropertyTextAttribute(true)]
[CategoryAttribute(L"OracleEBS")]
[ConfigurationPropertyAttribute(L"oraclePassword")]
public:
virtual property String^ OraclePassword {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OraclePassword ()

/** @property */
public final void set_OraclePassword (String value)
```

JScript

```
public final function get OraclePassword () : String

public final function set OraclePassword (value : String)
```

Property Value

The Oracle password.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.OracleUserName Property

Gets or sets the Oracle user name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("oracleUserName")> _
<CategoryAttribute("OracleEBS")> _
Public Property OracleUserName As String
```

C#

```
[ConfigurationPropertyAttribute("oracleUserName")]
[CategoryAttribute("OracleEBS")]
public string OracleUserName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleUserName")]
[CategoryAttribute(L"OracleEBS")]
public:
virtual property String^ OracleUserName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleUserName ()

/** @property */
public final void set_OracleUserName (String value)
```

JScript

```
public final function get OracleUserName () : String
public final function set OracleUserName (value : String)
```

Property Value

The Oracle user name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PolledDataAvailableStatement Property

Gets or sets the polled data available statement.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling")> _
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("polledDataAvailableStatement")> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("Polling")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("polledDataAvailableStatement")]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"polledDataAvailableStatement")]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PolledDataAvailableStatement ()

/** @property */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
public final function get PolledDataAvailableStatement () : String

public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The polled data available statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PollingAction Property

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
<ConfigurationPropertyAttribute("pollingAction")> _
Public Property PollingAction As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
[ConfigurationPropertyAttribute("pollingAction")]
public string PollingAction { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
[ConfigurationPropertyAttribute(L"pollingAction")]
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingAction ()

/** @property */
public final void set_PollingAction (String value)
```

JScript

```
public final function get PollingAction () : String

public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PollingInput Property

Gets or sets the polling input.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
<ConfigurationPropertyAttribute("pollingInput")> _
Public Property PollingInput As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
[ConfigurationPropertyAttribute("pollingInput")]
public string PollingInput { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
[ConfigurationPropertyAttribute(L"pollingInput")]
public:
virtual property String^ PollingInput {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingInput ()

/** @property */
public final void set_PollingInput (String value)
```

JScript

```
public final function get PollingInput () : String

public final function set PollingInput (value : String)
```

Property Value

The polling input.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PollingInterval Property

Gets or sets the transacted polling interval, that is, the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
<ConfigurationPropertyAttribute("pollingInterval")> _
Public Property PollingInterval As Integer
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
[ConfigurationPropertyAttribute("pollingInterval")]
public int PollingInterval { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
[ConfigurationPropertyAttribute(L"pollingInterval")]
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int

public final function set PollingInterval (value : int)
```

Property Value

The transacted polling interval.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PollWhileDataFound Property

Gets or sets whether the specified object continuously poll data when it is available.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling")> _
<ConfigurationPropertyAttribute("pollWhileDataFound")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling")]
[ConfigurationPropertyAttribute("pollWhileDataFound")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling")]
[ConfigurationPropertyAttribute(L"pollWhileDataFound")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_PollWhileDataFound ()

/** @property */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
public final function get PollWhileDataFound () : boolean

public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the specified object continuously poll data when it is available; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.PostPollStatement Property

Gets or sets the PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling")> _
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("postPollStatement")> _
Public Property PostPollStatement As String
```

C#

```
[CategoryAttribute("Polling")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("postPollStatement")]
public string PostPollStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"postPollStatement")]
public:
virtual property String^ PostPollStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PostPollStatement ()

/** @property */
public final void set_PostPollStatement (String value)
```

JScript

```
public final function get PostPollStatement () : String
public final function set PostPollStatement (value : String)
```

Property Value

The PL/SQL block that is executed after the polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.SkipNilNodes Property

Gets or sets whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("skipNilNodes")> _
<CategoryAttribute("Run Time Behavior")> _
Public Property SkipNilNodes As Boolean
```

C#

```
[ConfigurationPropertyAttribute("skipNilNodes")]
[CategoryAttribute("Run Time Behavior")]
public bool SkipNilNodes { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"skipNilNodes")]
[CategoryAttribute(L"Run Time Behavior")]
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_SkipNilNodes ()

/** @property */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
public final function get SkipNilNodes () : boolean

public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the nil node is skipped; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.StatementCachePurge Property

Gets or sets whether the data provider statement cache associated when a connection is purged.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("statementCachePurge")> _
Public Property StatementCachePurge As Boolean
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("statementCachePurge")]
public bool StatementCachePurge { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"statementCachePurge")]
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_StatementCachePurge ()

/** @property */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
public final function get StatementCachePurge () : boolean

public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the data provider statement cache associated with a connection is purged; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.StatementCacheSize Property

Gets or sets the maximum number of statements that can be cached by each data provider connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("BufferManagement")> _
<ConfigurationPropertyAttribute("statementCacheSize")> _
Public Property StatementCacheSize As Integer
```

C#

```
[CategoryAttribute("BufferManagement")]
[ConfigurationPropertyAttribute("statementCacheSize")]
public int StatementCacheSize { get; set; }
```

C++

```
[CategoryAttribute(L"BufferManagement")]
[ConfigurationPropertyAttribute(L"statementCacheSize")]
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int
public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements that can be cached by each data provider connection.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.UseAmbientTransaction Property

Gets or sets whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("useAmbientTransaction")> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useAmbientTransaction")]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useAmbientTransaction")]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the Oracle Database adapter performs the operations using the transaction context provided by the caller; otherwise, <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web

Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.UseOracleConnectionPool Property

Gets or sets whether to use the Oracle connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("useOracleConnectionPool")> _
<CategoryAttribute("OracleConnectionPool")> _
Public Property UseOracleConnectionPool As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useOracleConnectionPool")]
[CategoryAttribute("OracleConnectionPool")]
public bool UseOracleConnectionPool { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useOracleConnectionPool")]
[CategoryAttribute(L"OracleConnectionPool")]
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseOracleConnectionPool ()

/** @property */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
public final function get UseOracleConnectionPool () : boolean

public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if use the Oracle connection pool; otherwise,

<languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.UserAssembliesLoadPath Property

Gets or sets the path where the assemblies will be loaded.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("UDT .NET Type Generation - Run Time")> _
<ConfigurationPropertyAttribute("userAssembliesLoadPath")> _
Public Property UserAssembliesLoadPath As String
```

C#

```
[CategoryAttribute("UDT .NET Type Generation - Run Time")]
[ConfigurationPropertyAttribute("userAssembliesLoadPath")]
public string UserAssembliesLoadPath { get; set; }
```

C++

```
[CategoryAttribute(L"UDT .NET Type Generation - Run Time")]
[ConfigurationPropertyAttribute(L"userAssembliesLoadPath")]
public:
virtual property String^ UserAssembliesLoadPath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_UserAssembliesLoadPath ()

/** @property */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
public final function get UserAssembliesLoadPath () : String

public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The path where the assemblies will be loaded.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingConfigurationElement.UseSchemaInNameSpace Property

Gets or sets whether the schema name is included in the xml namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Metadata")> _
<ConfigurationPropertyAttribute("useSchemaInNameSpace")> _
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
[CategoryAttribute("Metadata")]
[ConfigurationPropertyAttribute("useSchemaInNameSpace")]
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
[CategoryAttribute(L"Metadata")]
[ConfigurationPropertyAttribute(L"useSchemaInNameSpace")]
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
public final function get UseSchemaInNameSpace () : boolean

public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>True</languageKeyword> if the schema name is included in the xml namespace for operations and their associated types; otherwise, <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows

Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingConfigurationElement Class](#)

[OracleEBSBindingConfigurationElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement Class

Enables the configuration of the Oracle E-Business Suite Adapter Binding from an application configuration file.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public Class OracleEBSBindingElementExtensionElement
    Inherits BindingElementExtensionElement
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public class OracleEBSBindingElementExtensionElement : BindingElementExtensionElement
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSBindingElementExtensionElement : public BindingElementExtensionElement
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public class OracleEBSBindingElementExtensionElement extends BindingElementExtensionElement
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public class OracleEBSBindingElementExtensionElement extends BindingElementExtensionElement
```

Remarks

The Microsoft BizTalk Adapter for Oracle E-Business Suite surfaces several binding properties. By setting these properties, you can control some of the adapter's behavior while interfacing with Oracle E-Business Suite or with the Oracle database.

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BindingElementExtensionElement](#)

Microsoft.Adapters.OracleEBS.OracleEBSBindingElementExtensionElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement Members

Enables the configuration of the Oracle E-Business Suite Adapter Binding from an application configuration file.

The following tables list the members exposed by the [OracleEBSBindingElementExtensionElement](#) type.

Public Constructors

Name	Description
OracleEBSBindingElementExtensionElement	Initializes a new instance of the OracleEBSBindingElementExtensionElement class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets a value that indicates whether the Oracle E-Business Suite connection URI is allowed to contain credentials.
 ApplicationShortName	Gets or sets the short name for an Oracle E-Business Suite application.
 BindingElementType	Overridden. Gets the binding element type.
 ClientCredentialType	Gets or sets the sets of credentials to specify when establishing a connection to Oracle E-Business Suite. The possible values are Database or EBusiness.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes the ODP.NET fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value that indicates how the adapter generates schema for a DataSet.
 EnablePerformanceCounters	Gets or sets a value that indicates whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets a value that indicates whether to enable or disable safe typing, that is, how the adapter surfaces certain Oracle data types.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the manner how the adapter receives messages from Oracle E-Business Suite. The possible values are Polling or Notification.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in an ODP.NET connection pool.
 MetadataPooling	Gets or sets a value that indicates whether ODP.NET caches metadata information for executed queries.

 MinPoolSize	Gets or sets the minimum number of connections in an ODP.NET connection pool.
 MlsSettings	Gets or sets the settings for multi-language support (MLS) specific options.
 NotificationPort	Gets or sets the port number that ODP.NET must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.
 NotifyOnListenerStart	Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OracleConnectionClientId	Gets or sets the client identifier (ID) for the connection used by the adapter to connect to Oracle.
 OracleEBSOrganizationId	Gets or sets the organization ID to which an Oracle E-Business Suite application belongs.
 OracleEBSResponsibilityKey	Gets or sets the responsibility key associated with the Oracle E-Business Suite user.
 OracleEBSResponsibilityName	Gets or sets the name of the responsibility associated with the Oracle E-Business Suite user.
 OraclePassword	Gets or sets the password credential for an Oracle database user or an Oracle E-Business Suite user.
 OracleUserName	Gets or sets the user name credential for an Oracle database user or an Oracle E-Business Suite user.
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling statement.
 PollingInterval	Gets or sets the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.
 PostPollStatement	Gets or sets a PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 SkipNilNodes	Gets or sets a value that indicates whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets a value that indicates whether the ODP.NET statement cache associated with a connection is purged when the connection is returned to the connection pool.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each ODP.NET connection.
 UseAmbientTransaction	Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets a value that indicates whether to use the ODP.NET connection pool.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets a value that indicates whether the schema name is included in the XML namespace for operations and their associated types.

[Top](#)

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	Overridden. Gets the collection of configuration properties.

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	Overridden. Applies the configuration properties from the specified binding element to the adapter.
	CopyFrom	Overridden. Copies the configuration properties from the specified configuration element.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	CreateBindingElement	Overridden. Creates the adapter binding element for Oracle E-Business Suite.
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	Overridden. Initializes the adapter with the configuration properties with the specified binding element.
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement Constructor

Initializes a new instance of the [OracleEBSBindingElementExtensionElement](#) class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSBindingElementExtensionElement ()
```

C++

```
public:  
OracleEBSBindingElementExtensionElement ()
```

J#

```
public OracleEBSBindingElementExtensionElement ()
```

JScript

```
public function OracleEBSBindingElementExtensionElement ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	Overridden. Applies the configuration properties from the specified binding element to the adapter.
	CopyFrom	Overridden. Copies the configuration properties from the specified configuration element.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBindingElement	Overridden. Creates the adapter binding element for Oracle E-Business Suite.
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	Overridden. Initializes the adapter with the configuration properties with the specified binding element.
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.ApplyConfiguration Method

Applies the configuration properties from the specified binding element to the adapter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub ApplyConfiguration ( _  
    bindingElement As BindingElement _  
)
```

C#

```
public override void ApplyConfiguration (  
    BindingElement bindingElement  
)
```

C++

```
public:  
virtual void ApplyConfiguration (  
    BindingElement^ bindingElement  
) override
```

J#

```
public void ApplyConfiguration (  
    BindingElement bindingElement  
)
```

JScript

```
public override function ApplyConfiguration (  
    bindingElement : BindingElement  
)
```

Parameters

bindingElement

The specified binding element.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.CopyFrom Method

Copies the configuration properties from the specified configuration element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

The specified configuration element.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.CreateBindingElement Method

Creates the adapter binding element for Oracle E-Business Suite.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Protected Overrides Function CreateBindingElement As BindingElement
```

C#

```
protected override BindingElement CreateBindingElement ()
```

C++

```
protected:  
virtual BindingElement^ CreateBindingElement () override
```

J#

```
protected BindingElement CreateBindingElement ()
```

JScript

```
protected override function CreateBindingElement () : BindingElement
```

Return Value

The adapter binding element for Oracle E-Business Suite.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.InitializeFrom Method

Initializes the adapter with the configuration properties with the specified binding element.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Protected Overrides Sub InitializeFrom ( _  
    bindingElement As BindingElement _  
)
```

C#

```
protected override void InitializeFrom (  
    BindingElement bindingElement  
)
```

C++

```
protected:  
virtual void InitializeFrom (  
    BindingElement^ bindingElement  
) override
```

J#

```
protected void InitializeFrom (  
    BindingElement bindingElement  
)
```

JScript

```
protected override function InitializeFrom (  
    bindingElement : BindingElement  
)
```

Parameters

bindingElement

The specified binding element.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	Gets or sets a value that indicates whether the Oracle E-Business Suite connection URI is allowed to contain credentials.
 ApplicationShortName	Gets or sets the short name for an Oracle E-Business Suite application.
 BindingElementType	Overridden. Gets the binding element type.
 ClientCredentialType	Gets or sets the sets of credentials to specify when establishing a connection to Oracle E-Business Suite. The possible values are Database or EBusiness.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ConnectionLifetime	Gets or sets the maximum duration in seconds of a connection.
 DataFetchSize	Gets or sets the amount of data in bytes the ODP.NET fetches from the result set in one server round trip.
 DecrPoolSize	Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value that indicates how the adapter generates schema for a DataSet.
 EnablePerformanceCounters	Gets or sets a value that indicates whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.
 EnableSafeTyping	Gets or sets a value that indicates whether to enable or disable safe typing, that is, how the adapter surfaces certain Oracle data types.
 GeneratedUserTypesAssemblyFilePath	Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.
 GeneratedUserTypesAssemblyKeyFilePath	Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.
 InboundOperationType	Gets or sets the manner how the adapter receives messages from Oracle E-Business Suite. The possible values are Polling or Notification.
 IncrPoolSize	Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool.
 InsertBatchSize	Gets or sets the batch size for multiple record Insert operations.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LongDatatypeColumnSize	Gets or sets the maximum size in bytes of an Oracle long data type column. Obsolete.
 MaxOutputAssociativeArrayElements	Gets or sets the size of the associative array that the adapter creates when performing operations that return an associative array in the response.
 MaxPoolSize	Gets or sets the maximum number of connections in an ODP.NET connection pool.
 MetadataPooling	Gets or sets a value that indicates whether ODP.NET caches metadata information for executed queries.
 MinPoolSize	Gets or sets the minimum number of connections in an ODP.NET connection pool.
 MlsSettings	Gets or sets the settings for multi-language support (MLS) specific options.
 NotificationPort	Gets or sets the port number that ODP.NET must open to listen for database change notification from Oracle database.
 NotificationStatement	Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

 NotifyOnListenerStart	Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.
 OracleConnectionClientId	Gets or sets the client identifier (ID) for the connection used by the adapter to connect to Oracle.
 OracleEBSOrganizationId	Gets or sets the organization ID to which an Oracle E-Business Suite application belongs.
 OracleEBSResponsibilityKey	Gets or sets the responsibility key associated with the Oracle E-Business Suite user.
 OracleEBSResponsibilityName	Gets or sets the name of the responsibility associated with the Oracle E-Business Suite user.
 OraclePassword	Gets or sets the password credential for an Oracle database user or an Oracle E-Business Suite user.
 OracleUserName	Gets or sets the user name credential for an Oracle database user or an Oracle E-Business Suite user.
 PolledDataAvailableStatement	Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.
 PollingAction	Gets or sets the action for the polling operation.
 PollingInput	Gets or sets the polling statement.
 PollingInterval	Gets or sets the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.
 PollWhileDataFound	Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.
 PostPollStatement	Gets or sets a PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.
 SkipNilNodes	Gets or sets a value that indicates whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.
 StatementCachePurge	Gets or sets a value that indicates whether the ODP.NET statement cache associated with a connection is purged when the connection is returned to the connection pool.
 StatementCacheSize	Gets or sets the maximum number of statements that can be cached by each ODP.NET connection.
 UseAmbientTransaction	Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.
 UseOracleConnectionPool	Gets or sets a value that indicates whether to use the ODP.NET connection pool.
 UserAssembliesLoadPath	Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.
 UseSchemaInNameSpace	Gets or sets a value that indicates whether the schema name is included in the XML namespace for operations and their associated types.

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	Overridden. Gets the collection of configuration properties.

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.AcceptCredentialsInUri Property

Gets or sets a value that indicates whether the Oracle E-Business Suite connection URI is allowed to contain credentials.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("acceptCredentialsInUri", DefaultValue:=False)> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[ConfigurationPropertyAttribute("acceptCredentialsInUri", DefaultValue=false)]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"acceptCredentialsInUri", DefaultValue=false)]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Property Value

<languageKeyword>>true</languageKeyword> if the Oracle E-Business Suite connection URI is allowed to contain credentials, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>>false</languageKeyword>.

Remarks

By default, the Oracle Database adapter throws an exception when the Oracle database credentials are specified in the connection URI. This is because these credentials are represented as plain text in the connection URI, and this poses a security risk.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.ApplicationShortName Property

Gets or sets the short name for an Oracle E-Business Suite application.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("applicationShortName", DefaultValue:=Nothing)> _
Public Property ApplicationShortName As String
```

C#

```
[ConfigurationPropertyAttribute("applicationShortName", DefaultValue=null)]
public string ApplicationShortName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"applicationShortName", DefaultValue=nullptr)]
public:
virtual property String^ ApplicationShortName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_ApplicationShortName ()

/** @property */
public final void set_ApplicationShortName (String value)
```

JScript

```
public final function get ApplicationShortName () : String

public final function set ApplicationShortName (value : String)
```

Property Value

The short name for an Oracle E-Business Suite application.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows

Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.BindingElementType Property

Gets the binding element type.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Overrides ReadOnly Property BindingElementType As Type
```

C#

```
public override Type BindingElementType { get; }
```

C++

```
public:  
virtual property Type^ BindingElementType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_BindingElementType ()
```

JScript

```
public override function get BindingElementType () : Type
```

Property Value

The binding element type.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

OracleEBSBindingElementExtensionElement.ClientCredentialType Property

Gets or sets the sets of credentials to specify when establishing a connection to Oracle E-Business Suite. The possible values are Database or EBusiness.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("clientCredentialType", DefaultValue:=ClientCredential.Database) > _
Public Property ClientCredentialType As ClientCredential
```

C#

```
[ConfigurationPropertyAttribute("clientCredentialType", DefaultValue=ClientCredential.Database)]
public ClientCredential ClientCredentialType { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"clientCredentialType", DefaultValue=ClientCredential::Database)]
public:
virtual property ClientCredential ClientCredentialType {
    ClientCredential get () sealed;
    void set (ClientCredential value) sealed;
}
```

J#

```
/** @property */
public final ClientCredential get_ClientCredentialType ()

/** @property */
public final void set_ClientCredentialType (ClientCredential value)
```

JScript

```
public final function get ClientCredentialType () : ClientCredential

public final function set ClientCredentialType (value : ClientCredential)
```

Property Value

The sets of credentials to specify when establishing a connection to Oracle E-Business Suite. The default is Database.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows

Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.ConnectionLifetime Property

Gets or sets the maximum duration in seconds of a connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("connectionLifetime", DefaultValue:=0)> _
Public Property ConnectionLifetime As Integer
```

C#

```
[ConfigurationPropertyAttribute("connectionLifetime", DefaultValue=0)]
public int ConnectionLifetime { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"connectionLifetime", DefaultValue=0)]
public:
virtual property int ConnectionLifetime {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ConnectionLifetime ()

/** @property */
public final void set_ConnectionLifetime (int value)
```

JScript

```
public final function get ConnectionLifetime () : int

public final function set ConnectionLifetime (value : int)
```

Property Value

The maximum duration in seconds of a connection. The default is zero.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.DataFetchSize Property

Gets or sets the amount of data in bytes the ODP.NET fetches from the result set in one server round trip.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("dataFetchSize", DefaultValue:=65536)> _
Public Property DataFetchSize As Long
```

C#

```
[ConfigurationPropertyAttribute("dataFetchSize", DefaultValue=65536)]
public long DataFetchSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"dataFetchSize", DefaultValue=65536)]
public:
virtual property long long DataFetchSize {
    long long get () sealed;
    void set (long long value) sealed;
}
```

J#

```
/** @property */
public final long get_DataFetchSize ()

/** @property */
public final void set_DataFetchSize (long value)
```

JScript

```
public final function get DataFetchSize () : long

public final function set DataFetchSize (value : long)
```

Property Value

The amount of data in bytes the ODP.NET fetches from the result set in one server round trip. The default is 65536.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.DecrPoolSize Property

Gets or sets the number of connections that are closed when an excessive amount of established connections are not in use.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("decrPoolSize", DefaultValue:=1)> _
Public Property DecrPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("decrPoolSize", DefaultValue=1)]
public int DecrPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"decrPoolSize", DefaultValue=1)]
public:
virtual property int DecrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_DecrPoolSize ()

/** @property */
public final void set_DecrPoolSize (int value)
```

JScript

```
public final function get DecrPoolSize () : int

public final function set DecrPoolSize (value : int)
```

Property Value

The number of connections that are closed when an excessive amount of established connections are not in use. The default is one.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.EnableBizTalkCompatibilityMode Property

Gets or sets a value that indicates how the adapter generates schema for a DataSet.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue:=True)> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue=true)]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableBizTalkCompatibilityMode", DefaultValue=true)]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean

public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if using the adapters from Biz Talk Server, otherwise <languageKeyword>>false</languageKeyword>.

Remarks

When using the adapters from BizTalk Server, you must always set the property to True. When using the adapters from Visual Studio, you must always set the property to False.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

OracleEBSBindingElementExtensionElement.EnablePerformanceCounters Property

Gets or sets a value that indicates whether to enable the WCF LOB Adapter SDK performance counters and the Oracle Database adapter LOB Latency performance counter.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue:=False)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue=false)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enablePerformanceCounters", DefaultValue=false)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

<languageKeyword>>true</languageKeyword> if performance counters is enabled, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.EnableSafeTyping Property

Gets or sets a value that indicates whether to enable or disable safe typing, that is, how the adapter surfaces certain Oracle data types.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enableSafeTyping", DefaultValue:=False)> _
Public Property EnableSafeTyping As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableSafeTyping", DefaultValue=false)]
public bool EnableSafeTyping { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableSafeTyping", DefaultValue=false)]
public:
virtual property bool EnableSafeTyping {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableSafeTyping ()

/** @property */
public final void set_EnableSafeTyping (boolean value)
```

JScript

```
public final function get EnableSafeTyping () : boolean

public final function set EnableSafeTyping (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if the safe typing is enabled, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.GeneratedUserTypesAssemblyFilePath Property

Gets or sets the name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath", DefaultValue:=Nothing)> _
<BindingPropertyDirectionAttribute(False, False)> _
Public Property GeneratedUserTypesAssemblyFilePath As String
```

C#

```
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyFilePath", DefaultValue=null)]
[BindingPropertyDirectionAttribute(false, false)]
public string GeneratedUserTypesAssemblyFilePath { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyFilePath", DefaultValue=nullptr)]
[BindingPropertyDirectionAttribute(false, false)]
public:
virtual property String^ GeneratedUserTypesAssemblyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyFilePath () : String
public final function set GeneratedUserTypesAssemblyFilePath (value : String)
```

Property Value

The name and path of the DLL that the adapter generates, while generating metadata, containing all UDTs that are used in the metadata.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.GeneratedUserTypesAssemblyKeyFilePath Property

Gets or sets the name and path of the key file that the adapter uses to create a strongly-typed assembly.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<BindingPropertyDirectionAttribute(False, False)> _
<ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath", DefaultValue:=Nothing)> _
Public Property GeneratedUserTypesAssemblyKeyFilePath As String
```

C#

```
[BindingPropertyDirectionAttribute(false, false)]
[ConfigurationPropertyAttribute("generatedUserTypesAssemblyKeyFilePath", DefaultValue=null)]
public string GeneratedUserTypesAssemblyKeyFilePath { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, false)]
[ConfigurationPropertyAttribute(L"generatedUserTypesAssemblyKeyFilePath", DefaultValue=nullptr)]
public:
virtual property String^ GeneratedUserTypesAssemblyKeyFilePath {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_GeneratedUserTypesAssemblyKeyFilePath ()

/** @property */
public final void set_GeneratedUserTypesAssemblyKeyFilePath (String value)
```

JScript

```
public final function get GeneratedUserTypesAssemblyKeyFilePath () : String
public final function set GeneratedUserTypesAssemblyKeyFilePath (value : String)
```

Property Value

The name and path of the key file that the adapter uses to create a strongly-typed assembly.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.InboundOperationType Property

Gets or sets the manner how the adapter receives messages from Oracle E-Business Suite. The possible values are Polling or Notification.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("inboundOperationType", DefaultValue:=InboundOperation.Polling)> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[ConfigurationPropertyAttribute("inboundOperationType", DefaultValue=InboundOperation.Polling)]
[BindingPropertyDirectionAttribute(true, false)]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"inboundOperationType", DefaultValue=InboundOperation::Polling)]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
public final InboundOperation get_InboundOperationType ()

/** @property */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
public final function get InboundOperationType () : InboundOperation
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

One of the enumeration values of [InboundOperation](#). The default is Polling.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP

Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.IncrPoolSize Property

Gets or sets the number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("incrPoolSize", DefaultValue:=5)> _
Public Property IncrPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("incrPoolSize", DefaultValue=5)]
public int IncrPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"incrPoolSize", DefaultValue=5)]
public:
virtual property int IncrPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_IncrPoolSize ()

/** @property */
public final void set_IncrPoolSize (int value)
```

JScript

```
public final function get IncrPoolSize () : int

public final function set IncrPoolSize (value : int)
```

Property Value

The number of new connections to be created when a new connection is requested and there are no available connections in the ODP.NET connection pool. The default is five.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.InsertBatchSize Property

Gets or sets the batch size for multiple record Insert operations.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("insertBatchSize", DefaultValue:=20)> _
Public Property InsertBatchSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("insertBatchSize", DefaultValue=20)]
public int InsertBatchSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"insertBatchSize", DefaultValue=20)]
public:
virtual property int InsertBatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_InsertBatchSize ()

/** @property */
public final void set_InsertBatchSize (int value)
```

JScript

```
public final function get InsertBatchSize () : int
public final function set InsertBatchSize (value : int)
```

Property Value

The batch size for multiple record Insert operations. The default is 20.

Remarks

The Oracle E-Business adapter batches the specified number of records into a single ODP.NET call. A well-chosen value for InsertBatchSize can greatly improve adapter performance for multiple record Insert operations.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.LongDatatypeColumnSize Property

NOTE: This property is now obsolete.

Gets or sets the maximum size in bytes of an Oracle long data type column.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")> _
<ConfigurationPropertyAttribute("longDatatypeColumnSize", DefaultValue:=0)> _
Public Property LongDatatypeColumnSize As Integer
```

C#

```
[ObsoleteAttribute("Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")]
[ConfigurationPropertyAttribute("longDatatypeColumnSize", DefaultValue=0)]
public int LongDatatypeColumnSize { get; set; }
```

C++

```
[ObsoleteAttribute(L"Long data type is deprecated in Oracle. Refer to Oracle documentation for suggested alternatives")]
[ConfigurationPropertyAttribute(L"longDatatypeColumnSize", DefaultValue=0)]
public:
virtual property int LongDatatypeColumnSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_LongDatatypeColumnSize ()

/** @property */
public final void set_LongDatatypeColumnSize (int value)
```

JScript

```
public final function get LongDatatypeColumnSize () : int
public final function set LongDatatypeColumnSize (value : int)
```

Property Value

The maximum size in bytes of an Oracle long data type column.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and

64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.MaxOutputAssociativeArrayElements Property

Gets or sets the size of the associate array that the adapter creates when performing operations that return an associative array in the response.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements", DefaultValue:=32)> _
Public Property MaxOutputAssociativeArrayElements As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxOutputAssociativeArrayElements", DefaultValue=32)]
public int MaxOutputAssociativeArrayElements { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxOutputAssociativeArrayElements", DefaultValue=32)]
public:
virtual property int MaxOutputAssociativeArrayElements {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxOutputAssociativeArrayElements ()

/** @property */
public final void set_MaxOutputAssociativeArrayElements (int value)
```

JScript

```
public final function get MaxOutputAssociativeArrayElements () : int

public final function set MaxOutputAssociativeArrayElements (value : int)
```

Property Value

The size of the associate array. The default is 32.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.MaxPoolSize Property

Gets or sets the maximum number of connections in an ODP.NET connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("maxPoolSize", DefaultValue:=100)> _
Public Property MaxPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxPoolSize", DefaultValue=100)]
public int MaxPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxPoolSize", DefaultValue=100)]
public:
virtual property int MaxPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxPoolSize ()

/** @property */
public final void set_MaxPoolSize (int value)
```

JScript

```
public final function get MaxPoolSize () : int

public final function set MaxPoolSize (value : int)
```

Property Value

The maximum number of connections in an ODP.NET connection pool. The default is 100.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.MetadataPooling Property

Gets or sets a value that indicates whether ODP.NET caches metadata information for executed queries.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("metadataPooling", DefaultValue:=True)> _
Public Property MetadataPooling As Boolean
```

C#

```
[ConfigurationPropertyAttribute("metadataPooling", DefaultValue=true)]
public bool MetadataPooling { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"metadataPooling", DefaultValue=true)]
public:
virtual property bool MetadataPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_MetadataPooling ()

/** @property */
public final void set_MetadataPooling (boolean value)
```

JScript

```
public final function get MetadataPooling () : boolean

public final function set MetadataPooling (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> when metadata pooling is enabled, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>true</languageKeyword>.

Remarks

Caching this information improves performance; however, if changes to the underlying Oracle artifacts occur on the Oracle system, this pooled metadata will be out of sync. This might cause operations performed on the Oracle system to return unexpected exceptions.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.MinPoolSize Property

Gets or sets the minimum number of connections in an ODP.NET connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
<ConfigurationPropertyAttribute("minPoolSize", DefaultValue:=1)> _  
Public Property MinPoolSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("minPoolSize", DefaultValue=1)]  
public int MinPoolSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"minPoolSize", DefaultValue=1)]  
public:  
virtual property int MinPoolSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_MinPoolSize ()  
  
/** @property */  
public final void set_MinPoolSize (int value)
```

JScript

```
public final function get MinPoolSize () : int  
public final function set MinPoolSize (value : int)
```

Property Value

The minimum number of connections in an ODP.NET connection pool. The default is one.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.MlsSettings Property

Gets or sets the settings for multi-language support (MLS) specific options.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("mIsSettings", DefaultValue:=Nothing)> _
Public Property MIsSettings As MIsSettings
```

C#

```
[ConfigurationPropertyAttribute("mIsSettings", DefaultValue=null)]
public MIsSettings MIsSettings { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"mIsSettings", DefaultValue=nullptr)]
public:
virtual property MIsSettings^ MIsSettings {
    MIsSettings^ get () sealed;
    void set (MIsSettings^ value) sealed;
}
```

J#

```
/** @property */
public final MIsSettings get_MIsSettings ()

/** @property */
public final void set_MIsSettings (MIsSettings value)
```

JScript

```
public final function get MIsSettings () : MIsSettings
public final function set MIsSettings (value : MIsSettings)
```

Property Value

The settings for MLS-specific options.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.NotificationPort Property

Gets or sets the port number that ODP.NET must open to listen for database change notification from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("notificationPort", DefaultValue:=-1)> _
<CategoryAttribute("Notification")> _
Public Property NotificationPort As Integer
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("notificationPort", DefaultValue=-1)]
[CategoryAttribute("Notification")]
public int NotificationPort { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"notificationPort", DefaultValue=-1)]
[CategoryAttribute(L"Notification")]
public:
virtual property int NotificationPort {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_NotificationPort ()

/** @property */
public final void set_NotificationPort (int value)
```

JScript

```
public final function get NotificationPort () : int

public final function set NotificationPort (value : int)
```

Property Value

The port number that ODP.NET must open to listen for database change notification from Oracle database. The default is -1, which signifies that ODP.NET uses a valid, random, unused port number.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.NotificationStatement Property

Gets or sets the SELECT statement used to register for getting notifications from Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("notificationStatement", DefaultValue:=Nothing)> _
<CategoryAttribute("Notification")> _
Public Property NotificationStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("notificationStatement", DefaultValue=null)]
[CategoryAttribute("Notification")]
public string NotificationStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"notificationStatement", DefaultValue=nullptr)]
[CategoryAttribute(L"Notification")]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NotificationStatement ()

/** @property */
public final void set_NotificationStatement (String value)
```

JScript

```
public final function get NotificationStatement () : String
public final function set NotificationStatement (value : String)
```

Property Value

The SELECT statement used to register for getting notifications from Oracle database.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.NotifyOnListenerStart Property

Gets or sets a value that indicates whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue:=True)> _
<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Notification")> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue=true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Notification")]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notifyOnListenerStart", DefaultValue=true)]
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Notification")]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NotifyOnListenerStart ()

/** @property */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
public final function get NotifyOnListenerStart () : boolean

public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if the adapter sends a notification to the adapter clients when listener starts, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>true</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.OracleConnectionClientId Property

Gets or sets the client identifier (ID) for the connection used by the adapter to connect to Oracle.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("oracleConnectionClientId", DefaultValue:=Nothing)> _
Public Property OracleConnectionClientId As String
```

C#

```
[ConfigurationPropertyAttribute("oracleConnectionClientId", DefaultValue=null)]
public string OracleConnectionClientId { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleConnectionClientId", DefaultValue=nullptr)]
public:
virtual property String^ OracleConnectionClientId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleConnectionClientId ()

/** @property */
public final void set_OracleConnectionClientId (String value)
```

JScript

```
public final function get OracleConnectionClientId () : String
public final function set OracleConnectionClientId (value : String)
```

Property Value

The client ID for the connection used by the adapter to connect to Oracle.

Remarks

This is useful if you have enabled Oracle server-side tracing, and want to selectively trace the operations for a particular client ID. This also allows you to filter and view the Oracle server traces based on a client ID.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.OracleEBSOrganizationId Property

Gets or sets the organization ID to which an Oracle E-Business Suite application belongs.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("oracleEBSOrganizationId", DefaultValue:=Nothing)> _
Public Property OracleEBSOrganizationId As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSOrganizationId", DefaultValue=null)]
public string OracleEBSOrganizationId { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSOrganizationId", DefaultValue=nullptr)]
public:
virtual property String^ OracleEBSOrganizationId {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSOrganizationId ()

/** @property */
public final void set_OracleEBSOrganizationId (String value)
```

JScript

```
public final function get OracleEBSOrganizationId () : String
public final function set OracleEBSOrganizationId (value : String)
```

Property Value

The organization ID to which an Oracle E-Business Suite application belongs.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional

with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.OracleEBSResponsibilityKey Property

Gets or sets the responsibility key associated with the Oracle E-Business Suite user.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("oracleEBSResponsibilityKey", DefaultValue:=Nothing)> _
Public Property OracleEBSResponsibilityKey As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSResponsibilityKey", DefaultValue=null)]
public string OracleEBSResponsibilityKey { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSResponsibilityKey", DefaultValue=nullptr)]
public:
virtual property String^ OracleEBSResponsibilityKey {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSResponsibilityKey ()

/** @property */
public final void set_OracleEBSResponsibilityKey (String value)
```

JScript

```
public final function get OracleEBSResponsibilityKey () : String
public final function set OracleEBSResponsibilityKey (value : String)
```

Property Value

The responsibility key associated with the Oracle E-Business Suite user.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)
[OracleEBSBindingElementExtensionElement Members](#)
[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.OracleEBSResponsibilityName Property

Gets or sets the name of the responsibility associated with the Oracle E-Business Suite user.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("oracleEBSResponsibilityName", DefaultValue:=Nothing)> _
Public Property OracleEBSResponsibilityName As String
```

C#

```
[ConfigurationPropertyAttribute("oracleEBSResponsibilityName", DefaultValue=null)]
public string OracleEBSResponsibilityName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleEBSResponsibilityName", DefaultValue=nullptr)]
public:
virtual property String^ OracleEBSResponsibilityName {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleEBSResponsibilityName ()

/** @property */
public final void set_OracleEBSResponsibilityName (String value)
```

JScript

```
public final function get OracleEBSResponsibilityName () : String

public final function set OracleEBSResponsibilityName (value : String)
```

Property Value

The name of the responsibility associated with the Oracle E-Business Suite user.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

OracleEBSBindingElementExtensionElement.OraclePassword Property

Gets or sets the password credential for an Oracle database user or an Oracle E-Business Suite user.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("oraclePassword", DefaultValue:=Nothing)> _
Public Property OraclePassword As String
```

C#

```
[ConfigurationPropertyAttribute("oraclePassword", DefaultValue=null)]
public string OraclePassword { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oraclePassword", DefaultValue=nullptr)]
public:
virtual property String^ OraclePassword {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OraclePassword ()

/** @property */
public final void set_OraclePassword (String value)
```

JScript

```
public final function get OraclePassword () : String

public final function set OraclePassword (value : String)
```

Property Value

The password credential for an Oracle database user or an Oracle E-Business Suite user.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.OracleUserName Property

Gets or sets the user name credential for an Oracle database user or an Oracle E-Business Suite user.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("oracleUserName", DefaultValue:=Nothing)> _
Public Property OracleUserName As String
```

C#

```
[ConfigurationPropertyAttribute("oracleUserName", DefaultValue=null)]
public string OracleUserName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"oracleUserName", DefaultValue=nullptr)]
public:
virtual property String^ OracleUserName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_OracleUserName ()

/** @property */
public final void set_OracleUserName (String value)
```

JScript

```
public final function get OracleUserName () : String

public final function set OracleUserName (value : String)
```

Property Value

The user name credential for an Oracle database user or an Oracle E-Business Suite user.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PolledDataAvailableStatement Property

Gets or sets the SELECT statement executed to determine whether any data is available for polling for a specific table.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue:=Nothing)> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue=null)]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"polledDataAvailableStatement", DefaultValue=nullptr)]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PolledDataAvailableStatement ()

/** @property */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
public final function get PolledDataAvailableStatement () : String

public final function set PolledDataAvailableStatement (value : String)
```

Property Value

The SELECT statement executed to determine whether any data is available for polling for a specific table.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)
[OracleEBSBindingElementExtensionElement Members](#)
[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PollingAction Property

Gets or sets the action for the polling operation.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingAction", DefaultValue:=Nothing)> _
Public Property PollingAction As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingAction", DefaultValue=null)]
public string PollingAction { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingAction", DefaultValue=nullptr)]
public:
virtual property String^ PollingAction {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingAction ()

/** @property */
public final void set_PollingAction (String value)
```

JScript

```
public final function get PollingAction () : String
public final function set PollingAction (value : String)
```

Property Value

The action for the polling operation.

Remarks

You can determine the polling action for a specific operation from the metadata you generate for the operation using the Consume Adapter Service Add-in.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PollingInput Property

Gets or sets the polling statement.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingInput", DefaultValue:=Nothing)> _
Public Property PollingInput As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingInput", DefaultValue=null)]
public string PollingInput { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingInput", DefaultValue=nullptr)]
public:
virtual property String^ PollingInput {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingInput ()

/** @property */
public final void set_PollingInput (String value)
```

JScript

```
public final function get PollingInput () : String
public final function set PollingInput (value : String)
```

Property Value

The polling statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PollingInterval Property

Gets or sets the interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingInterval", DefaultValue:=30)> _
Public Property PollingInterval As Integer
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingInterval", DefaultValue=30)]
public int PollingInterval { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingInterval", DefaultValue=30)]
public:
virtual property int PollingInterval {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingInterval ()

/** @property */
public final void set_PollingInterval (int value)
```

JScript

```
public final function get PollingInterval () : int
public final function set PollingInterval (value : int)
```

Property Value

The interval in seconds at which the Oracle Database adapter executes the polling statement against the Oracle database. The default is 30.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PollWhileDataFound Property

Gets or sets a value that indicates whether the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database, if data is available in the table being polled.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue:=False)> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue=false)]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollWhileDataFound", DefaultValue=false)]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_PollWhileDataFound ()

/** @property */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
public final function get PollWhileDataFound () : boolean

public final function set PollWhileDataFound (value : boolean)
```

Property Value

<languageKeyword>>true</languageKeyword> if the Oracle Database adapter ignores the polling interval and continuously polls the Oracle database while data is available; <languageKeyword>>false</languageKeyword> if the Oracle adapters executes the SQL Statement at the specified polling interval. The default is <languageKeyword>>false</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.PostPollStatement Property

Gets or sets a PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("postPollStatement", DefaultValue:=Nothing)> _
Public Property PostPollStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("postPollStatement", DefaultValue=null)]
public string PostPollStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"postPollStatement", DefaultValue=nullptr)]
public:
virtual property String^ PostPollStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PostPollStatement ()

/** @property */
public final void set_PostPollStatement (String value)
```

JScript

```
public final function get PostPollStatement () : String
public final function set PostPollStatement (value : String)
```

Property Value

A PL/SQL block that is executed after the polling statement and before the /POLLINGSTMT message is sent to the consumer. The default is <languageKeyword>null</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.Properties Property

Gets the collection of configuration properties.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Protected Overrides ReadOnly Property Properties As ConfigurationPropertyCollection
```

C#

```
protected override ConfigurationPropertyCollection Properties { get; }
```

C++

```
protected:  
virtual property ConfigurationPropertyCollection^ Properties {  
    ConfigurationPropertyCollection^ get () override;  
}
```

J#

```
/** @property */  
protected ConfigurationPropertyCollection get_Properties ()
```

JScript

```
protected override function get Properties () : ConfigurationPropertyCollection
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.SkipNilNodes Property

Gets or sets a value that indicates whether the Oracle Database adapter will skip inserting or updating values for nodes that are marked as ?nil? in the request XML.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("skipNilNodes", DefaultValue:=True)> _
<CategoryAttribute("Run Time Behavior")> _
Public Property SkipNilNodes As Boolean
```

C#

```
[ConfigurationPropertyAttribute("skipNilNodes", DefaultValue=true)]
[CategoryAttribute("Run Time Behavior")]
public bool SkipNilNodes { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"skipNilNodes", DefaultValue=true)]
[CategoryAttribute(L"Run Time Behavior")]
public:
virtual property bool SkipNilNodes {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_SkipNilNodes ()

/** @property */
public final void set_SkipNilNodes (boolean value)
```

JScript

```
public final function get SkipNilNodes () : boolean

public final function set SkipNilNodes (value : boolean)
```

Property Value

<languageKeyword>>true</languageKeyword> if the adapter will skip passing values for nodes that are marked as 'nil';
 <languageKeyword>>false</languageKeyword> if the adapter explicitly passes a null value for these nodes. The default is
 <languageKeyword>>false</languageKeyword>.

Remarks

This binding property is applicable for inserting or updating records in a table and for RECORD type parameters in stored procedures.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.StatementCachePurge Property

Gets or sets a value that indicates whether the ODP.NET statement cache associated with a connection is purged when the connection is returned to the connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("statementCachePurge", DefaultValue:=False)> _
Public Property StatementCachePurge As Boolean
```

C#

```
[ConfigurationPropertyAttribute("statementCachePurge", DefaultValue=false)]
public bool StatementCachePurge { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"statementCachePurge", DefaultValue=false)]
public:
virtual property bool StatementCachePurge {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_StatementCachePurge ()

/** @property */
public final void set_StatementCachePurge (boolean value)
```

JScript

```
public final function get StatementCachePurge () : boolean

public final function set StatementCachePurge (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> when the statement cache purging is enabled, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>>false</languageKeyword>.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of

Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.StatementCacheSize Property

Gets or sets the maximum number of statements that can be cached by each ODP.NET connection.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("statementCacheSize", DefaultValue:=10)> _
Public Property StatementCacheSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("statementCacheSize", DefaultValue=10)]
public int StatementCacheSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"statementCacheSize", DefaultValue=10)]
public:
virtual property int StatementCacheSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_StatementCacheSize ()

/** @property */
public final void set_StatementCacheSize (int value)
```

JScript

```
public final function get StatementCacheSize () : int

public final function set StatementCacheSize (value : int)
```

Property Value

The maximum number of statements that can be cached by each ODP.NET connection. The default is 10.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit

editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.UseAmbientTransaction Property

Gets or sets a value that indicates whether the Oracle Database adapter performs the operations using the transaction context provided by the caller.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue:=True)> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue=true)]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useAmbientTransaction", DefaultValue=true)]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if the adapter always performs the operations in a transaction context, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>true</languageKeyword>.

Remarks

Not performing operations in a transactional context is advisable only for operations that do not make changes to the database. For operations that update data in the database, we recommend setting the binding property to true otherwise you might either experience message loss or duplicate messages depending on whether you are performing inbound or outbound operations.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.UseOracleConnectionPool Property

Gets or sets a value that indicates whether to use the ODP.NET connection pool.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("useOracleConnectionPool", DefaultValue:=True)> _
Public Property UseOracleConnectionPool As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useOracleConnectionPool", DefaultValue=true)]
public bool UseOracleConnectionPool { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useOracleConnectionPool", DefaultValue=true)]
public:
virtual property bool UseOracleConnectionPool {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseOracleConnectionPool ()

/** @property */
public final void set_UseOracleConnectionPool (boolean value)
```

JScript

```
public final function get UseOracleConnectionPool () : boolean
public final function set UseOracleConnectionPool (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if the ODP.NET connection pool is used, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>true</languageKeyword>.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and

64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.UserAssembliesLoadPath Property

Gets or sets the name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("userAssembliesLoadPath", DefaultValue:=Nothing)> _
Public Property UserAssembliesLoadPath As String
```

C#

```
[ConfigurationPropertyAttribute("userAssembliesLoadPath", DefaultValue=null)]
public string UserAssembliesLoadPath { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"userAssembliesLoadPath", DefaultValue=nullptr)]
public:
virtual property String^ UserAssembliesLoadPath {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_UserAssembliesLoadPath ()

/** @property */
public final void set_UserAssembliesLoadPath (String value)
```

JScript

```
public final function get UserAssembliesLoadPath () : String

public final function set UserAssembliesLoadPath (value : String)
```

Property Value

The name of the DLLs, separated by a semi-colon, which the adapter creates while generating metadata.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional

with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSBindingElementExtensionElement.UseSchemaInNameSpace Property

Gets or sets a value that indicates whether the schema name is included in the XML namespace for operations and their associated types.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("useSchemaInNameSpace", DefaultValue:=True)> _
Public Property UseSchemaInNameSpace As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useSchemaInNameSpace", DefaultValue=true)]
public bool UseSchemaInNameSpace { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useSchemaInNameSpace", DefaultValue=true)]
public:
virtual property bool UseSchemaInNameSpace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseSchemaInNameSpace ()

/** @property */
public final void set_UseSchemaInNameSpace (boolean value)
```

JScript

```
public final function get UseSchemaInNameSpace () : boolean

public final function set UseSchemaInNameSpace (value : boolean)
```

Property Value

<languageKeyword>true</languageKeyword> if the schema name is included in the XML namespace, otherwise <languageKeyword>>false</languageKeyword>. The default is <languageKeyword>true</languageKeyword>.

Remarks

The advantage of not having scheme name included in the namespace is that if there is a table with same name in two different schemas then the same XML can be used to perform the simple SQL operations (Insert, Update, Delete, Select) on both tables.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and

64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSBindingElementExtensionElement Class](#)

[OracleEBSBindingElementExtensionElement Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Class

Builds the Uniform Resource Identifier (URI) for establishing connection to the Oracle E-Business Suite and essentially the underlying Oracle database.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public Class OracleEBSConnectionUri
    Inherits ConnectionUri
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public class OracleEBSConnectionUri : ConnectionUri
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSConnectionUri : public ConnectionUri
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public class OracleEBSConnectionUri extends ConnectionUri
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public class OracleEBSConnectionUri extends ConnectionUri
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.ServiceModel.Channels.Common.ConnectionUri](#)

Microsoft.Adapters.OracleEBS.OracleEBSConnectionUri

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Members

Builds the Uniform Resource Identifier (URI) for establishing connection to the Oracle E-Business Suite and essentially the underlying Oracle database.

The following tables list the members exposed by the [OracleEBSConnectionUri](#) type.

Public Constructors

Name	Description
OracleEBSConnectionUri	Overloaded. Initializes a new instance of the OracleEBSConnectionUri class.

Top

Public Properties

Name	Description
 DataSourceName	Gets or sets the Oracle data source name.
 PortNumber	Gets or sets the Oracle Net listener port.
 SampleUriString	Overridden. Gets formatted URI string.
 SecureUriString	(inherited from ConnectionUri)
 ServerAddress	Gets or sets the name or IP address of the server on which the Oracle E-Business Suite is running.
 ServiceName	Gets or sets the Oracle database service name.
 ServiceType	Gets or sets the Oracle service type. The possible values are Dedicated or Shared.
 Uri	Overridden. Gets or sets the connection URI.

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetAttributes	(inherited from ConnectionUri)
 GetClassName	(inherited from ConnectionUri)
 GetComponentName	(inherited from ConnectionUri)
 GetConverter	(inherited from ConnectionUri)
 GetDefaultEvent	(inherited from ConnectionUri)
 GetDefaultProperty	(inherited from ConnectionUri)
 GetEditor	(inherited from ConnectionUri)
 GetEvents	Overloaded. (inherited from ConnectionUri)
 GetHashCode	(inherited from Object)
 GetProperties	Overloaded. (inherited from ConnectionUri)
 GetPropertyOwner	(inherited from ConnectionUri)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from ConnectionUri)

Top

Protected Methods

Name	Description
 Finalize	(inherited from Object)
  GetQueryStringValue	(inherited from ConnectionUri)
  GetUserInfoValue	(inherited from ConnectionUri)
 MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[OracleEBSConnectionUri Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Constructor

Initializes a new instance of the [OracleEBSConnectionUri](#) class.

Overload List

Name	Description
OracleEBSConnectionUri ()	Initializes a new instance of the ConnectionUri class.
OracleEBSConnectionUri (Uri, Boolean)	Initializes a new instance of the ConnectionUri class with a URI object.

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Constructor ()

Initializes a new instance of the ConnectionUri class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSConnectionUri ()
```

C++

```
public:  
OracleEBSConnectionUri ()
```

J#

```
public OracleEBSConnectionUri ()
```

JScript

```
public function OracleEBSConnectionUri ()
```

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Constructor (Uri, Boolean)

Initializes a new instance of the ConnectionUri class with a URI object.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Sub New ( _
    uri As Uri, _
    enableCredentialsInUri As Boolean _
)
```

C#

```
public OracleEBSConnectionUri (
    Uri uri,
    bool enableCredentialsInUri
)
```

C++

```
public:
    OracleEBSConnectionUri (
        Uri^ uri,
        bool enableCredentialsInUri
    )
```

J#

```
public OracleEBSConnectionUri (
    Uri uri,
    boolean enableCredentialsInUri
)
```

JScript

```
public function OracleEBSConnectionUri (
    uri : Uri,
    enableCredentialsInUri : boolean
)
```

Parameters

uri

The connection URI.

enableCredentialsInUri

<languageKeyword>>true</languageKeyword> to enable credentials in connection URI, otherwise <languageKeyword>>false</languageKeyword> .

Remarks

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetAttributes	(inherited from ConnectionUri)
	GetClassName	(inherited from ConnectionUri)
	GetComponentName	(inherited from ConnectionUri)
	GetConverter	(inherited from ConnectionUri)
	GetDefaultEvent	(inherited from ConnectionUri)
	GetDefaultProperty	(inherited from ConnectionUri)
	GetEditor	(inherited from ConnectionUri)
	GetEvents	Overloaded. (inherited from ConnectionUri)
	GetHashCode	(inherited from Object)
	GetProperties	Overloaded. (inherited from ConnectionUri)
	GetPropertyOwner	(inherited from ConnectionUri)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from ConnectionUri)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	 GetQueryStringValue	(inherited from ConnectionUri)
	 GetUserInfoValue	(inherited from ConnectionUri)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OracleEBSConnectionUri Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri Properties

Public Properties

	Name	Description
	DataSourceName	Gets or sets the Oracle data source name.
	PortNumber	Gets or sets the Oracle Net listener port.
	SampleUriString	Overridden. Gets formatted URI string.
	SecureUriString	(inherited from ConnectionUri)
	ServerAddress	Gets or sets the name or IP address of the server on which the Oracle E-Business Suite is running.
	ServiceName	Gets or sets the Oracle database service name.
	ServiceType	Gets or sets the Oracle service type. The possible values are Dedicated or Shared.
	Uri	Overridden. Gets or sets the connection URI.

See Also

Reference

[OracleEBSConnectionUri Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.DataSourceName Property

Gets or sets the Oracle data source name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("TNS Alias")> _
Public Property DataSourceName As String
```

C#

```
[CategoryAttribute("TNS Alias")]
public string DataSourceName { get; set; }
```

C++

```
[CategoryAttribute(L"TNS Alias")]
public:
property String^ DataSourceName {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_DataSourceName ()

/** @property */
public void set_DataSourceName (String value)
```

JScript

```
public function get DataSourceName () : String

public function set DataSourceName (value : String)
```

Property Value

The Oracle data source name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.PortNumber Property

Gets or sets the Oracle Net listener port.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Connect Descriptor")> _
<DefaultValueAttribute(1521)> _
Public Property PortNumber As Integer
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Connect Descriptor")]
[DefaultValueAttribute(1521)]
public int PortNumber { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Connect Descriptor")]
[DefaultValueAttribute(1521)]
public:
property int PortNumber {
    int get ();
    void set (int value);
}
```

J#

```
/** @property */
public int get_PortNumber ()

/** @property */
public void set_PortNumber (int value)
```

JScript

```
public function get PortNumber () : int

public function set PortNumber (value : int)
```

Property Value

The Oracle Net listener port.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.SampleUriString Property

Gets formatted URI string.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SampleUriString As String
```

C#

```
public override string SampleUriString { get; }
```

C++

```
public:  
virtual property String^ SampleUriString {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_SampleUriString ()
```

JScript

```
public override function get SampleUriString () : String
```

Property Value

The formatted URI string.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.ServerAddress Property

Gets or sets the name or IP address of the server on which the Oracle E-Business Suite is running.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServerAddress As String
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public string ServerAddress { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property String^ ServerAddress {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public String get_ServerAddress ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServerAddress (String value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServerAddress () : String

CompilerGeneratedAttribute
public function set ServerAddress (value : String)
```

Property Value

The name or IP address of the server on which the Oracle E-Business Suite is running.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.ServiceName Property

Gets or sets the Oracle database service name.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServiceName As String
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public string ServiceName { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property String^ ServiceName {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public String get_ServiceName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServiceName (String value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServiceName () : String

CompilerGeneratedAttribute
public function set ServiceName (value : String)
```

Property Value

The Oracle database service name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.ServiceType Property

Gets or sets the Oracle service type. The possible values are Dedicated or Shared.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connect Descriptor")> _
Public Property ServiceType As OracleServiceType
```

C#

```
[CategoryAttribute("Connect Descriptor")]
public OracleServiceType ServiceType { get; set; }
```

C++

```
[CategoryAttribute(L"Connect Descriptor")]
public:
property OracleServiceType ServiceType {
OracleServiceType get ();
void set (OracleServiceType value);
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public OracleServiceType get_ServiceType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public void set_ServiceType (OracleServiceType value)
```

JScript

```
CompilerGeneratedAttribute
public function get ServiceType () : OracleServiceType

CompilerGeneratedAttribute
public function set ServiceType (value : OracleServiceType)
```

Property Value

One of the enumeration values of the [OracleServiceType](#). The default is Dedicated.

Remarks

A dedicated service uses a dedicated server process to serve only one user process. A shared service uses a shared server process that can serve multiple user processes.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSConnectionUri.Uri Property

Gets or sets the connection URI.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property Uri As Uri
```

C#

```
public override Uri Uri { get; set; }
```

C++

```
public:  
virtual property Uri^ Uri {  
    Uri^ get () override;  
    void set (Uri^ value) override;  
}
```

J#

```
/** @property */  
public Uri get Uri ()  
  
/** @property */  
public void set Uri (Uri value)
```

JScript

```
public override function get Uri () : Uri  
  
public override function set Uri (value : Uri)
```

Property Value

The connection URI.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSConnectionUri Class](#)

[OracleEBSConnectionUri Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior Class

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")> _
Public NotInheritable Class OracleEBSInboundTransactionBehavior
    Inherits BehaviorExtensionElement
    Implements IServiceBehavior
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")]
public sealed class OracleEBSInboundTransactionBehavior : BehaviorExtensionElement, IServiceBehavior
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1705:LongAcronymsShouldBePascalCased")]
public ref class OracleEBSInboundTransactionBehavior sealed : public BehaviorExtensionElement, IServiceBehavior
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased") */
public final class OracleEBSInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1705:LongAcronymsShouldBePascalCased")
public final class OracleEBSInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BehaviorExtensionElement](#)

Microsoft.Adapters.OracleEBS.OracleEBSInboundTransactionBehavior

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior Members

The following tables list the members exposed by the [OracleEBSInboundTransactionBehavior](#) type.

Public Constructors

Name	Description
OracleEBSInboundTransactionBehavior	

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BehaviorType	Overridden.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 TransactionIsolationLevel	
 TransactionTimeout	

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from ConfigurationElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 CopyFrom	Overridden.
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 CreateBehavior	(inherited from BehaviorExtensionElement)
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ServiceModelExtensionElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ServiceModelExtensionElement)

 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ServiceModelExtensionElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

[Top](#)

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior Constructor

Initializes a new instance of the [OracleEBSInboundTransactionBehavior](#) Class.

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OracleEBSInboundTransactionBehavior ()
```

C++

```
public:  
OracleEBSInboundTransactionBehavior ()
```

J#

```
public OracleEBSInboundTransactionBehavior ()
```

JScript

```
public function OracleEBSInboundTransactionBehavior ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	CopyFrom	Overridden.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBehavior	(inherited from BehaviorExtensionElement)
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.CopyFrom Method

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.AddBindingParameters Method

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

endpoints

bindingParameters

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior Method

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.Validate Method

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	BehaviorType	Overridden.
	ConfigurationElementName	(inherited from ServiceModelExtensionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	TransactionIsolationLevel	
	TransactionTimeout	

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from ConfigurationElement)

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.BehaviorType Property

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property BehaviorType As Type
```

C#

```
public override Type BehaviorType { get; }
```

C++

```
public:
virtual property Type^ BehaviorType {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_BehaviorType ()
```

JScript

```
public override function get BehaviorType () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

OracleEBSInboundTransactionBehavior.TransactionIsolationLevel Property

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue:=IsolationLevel.Serializable)> _
Public Property TransactionIsolationLevel As IsolationLevel
```

C#

```
[ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue=IsolationLevel.Serializable)]
public IsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionIsolationLevel", DefaultValue=IsolationLevel::Serializable)]
public:
property IsolationLevel TransactionIsolationLevel {
    IsolationLevel get ();
    void set (IsolationLevel value);
}
```

J#

```
/** @property */
public IsolationLevel get_TransactionIsolationLevel ()

/** @property */
public void set_TransactionIsolationLevel (IsolationLevel value)
```

JScript

```
public function get TransactionIsolationLevel () : IsolationLevel

public function set TransactionIsolationLevel (value : IsolationLevel)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleEBSInboundTransactionBehavior.TransactionTimeout Property

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("transactionTimeout")> _
Public Property TransactionTimeout As TimeSpan
```

C#

```
[ConfigurationPropertyAttribute("transactionTimeout")]
public TimeSpan TransactionTimeout { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionTimeout")]
public:
property TimeSpan TransactionTimeout {
    TimeSpan get ();
    void set (TimeSpan value);
}
```

J#

```
/** @property */
public TimeSpan get_TransactionTimeout ()

/** @property */
public void set_TransactionTimeout (TimeSpan value)
```

JScript

```
public function get TransactionTimeout () : TimeSpan

public function set TransactionTimeout (value : TimeSpan)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OracleEBSInboundTransactionBehavior Class](#)

[OracleEBSInboundTransactionBehavior Members](#)

[Microsoft.Adapters.OracleEBS Namespace](#)

OracleServiceType Enumeration

Namespace: Microsoft.Adapters.OracleEBS

Assembly: Microsoft.Adapters.OracleEBS (in microsoft.adapters.oracleebs.dll)

Syntax

VB

```
'Declaration
Public Enumeration OracleServiceType
```

C#

```
public enum OracleServiceType
```

C++

```
public enum class OracleServiceType
```

J#

```
public enum OracleServiceType
```

JScript

```
public enum OracleServiceType
```

Members

Member name	Description
Dedicated	
Shared	

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.OracleEBS Namespace](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace

Classes

Class	Description
ApplicationContextPropertySchema	
ApplicationShortName	
Language	
OrganizationId	
ResponsibilityKey	
ResponsibilityName	

ApplicationContextPropertySchema Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<SchemaTypeAttribute(SchemaTypeEnum.Property)> _
Public NotInheritable Class ApplicationContextPropertySchema
    Inherits SchemaBase
```

C#

```
[SerializableAttribute]
[SchemaTypeAttribute(SchemaTypeEnum.Property)]
public sealed class ApplicationContextPropertySchema : SchemaBase
```

C++

```
[SerializableAttribute]
[SchemaTypeAttribute(SchemaTypeEnum::Property)]
public ref class ApplicationContextPropertySchema sealed : public SchemaBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute SchemaTypeAttribute(SchemaTypeEnum.Property) */
public final class ApplicationContextPropertySchema extends SchemaBase
```

JScript

```
SerializableAttribute
SchemaTypeAttribute(SchemaTypeEnum.Property)
public final class ApplicationContextPropertySchema extends SchemaBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.SchemaBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ApplicationContextPropertySchema

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationContextPropertySchema Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationContextPropertySchema Members

The following tables list the members exposed by the [ApplicationContextPropertySchema](#) type.

Public Constructors

Name	Description
ApplicationContextPropertySchema	

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 ParseTables	(inherited from SchemaBase)
 RootNodes	Overridden.
 Schema	(inherited from SchemaBase)
 SchemaCollection	(inherited from SchemaBase)
 XmlContent	Overridden.

Top

Protected Properties

Name	Description
 RawSchema	(inherited from SchemaBase)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 CreateResolvedSchema	(inherited from SchemaBase)
 Equals	Overloaded. (inherited from Object)
  FindReferencedSchema	(inherited from SchemaBase)
  FindReferencedSchemaType	(inherited from SchemaBase)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationContextPropertySchema Constructor

Initializes a new instance of the [ApplicationContextPropertySchema](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public ApplicationContextPropertySchema ()
```

C++

```
public:  
ApplicationContextPropertySchema ()
```

J#

```
public ApplicationContextPropertySchema ()
```

JScript

```
public function ApplicationContextPropertySchema ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[ApplicationContextPropertySchema Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationContextPropertySchema Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	CreateResolvedSchema	(inherited from SchemaBase)
	Equals	Overloaded. (inherited from Object)
 	FindReferencedSchema	(inherited from SchemaBase)
 	FindReferencedSchemaType	(inherited from SchemaBase)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationContextPropertySchema Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	ParseTables	(inherited from SchemaBase)
	RootNodes	Overridden.
	Schema	(inherited from SchemaBase)
	SchemaCollection	(inherited from SchemaBase)
	XmlContent	Overridden.

Protected Properties

	Name	Description
	RawSchema	(inherited from SchemaBase)

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationContextPropertySchema.RootNodes Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration
Public Overrides ReadOnly Property RootNodes As String()
```

C#

```
public override string[] RootNodes { get; }
```

C++

```
public:
virtual property array<String^>^ RootNodes {
    array<String^>^ get () override;
}
```

J#

```
/** @property */
public String[] get_RootNodes ()
```

JScript

```
public override function get RootNodes () : String[]
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[ApplicationContextPropertySchema Members](#)

ApplicationContextPropertySchema.XmlContent Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property XmlContent As String
```

C#

```
public override string XmlContent { get; }
```

C++

```
public:  
virtual property String^ XmlContent {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_XmlContent ()
```

JScript

```
public override function get XmlContent () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationContextPropertySchema Class](#)

[ApplicationContextPropertySchema Members](#)

ApplicationShortName Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<IsSensitivePropertyAttribute(False)> _
<PropertyGuidAttribute("f77a3fcb-4204-41af-8176-a94757e41fe3")> _
<PropertyTypeAttribute("ApplicationShortName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")> _
Public NotInheritable Class ApplicationShortName
    Inherits MessageContextPropertyBase
```

C#

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyGuidAttribute("f77a3fcb-4204-41af-8176-a94757e41fe3")]
[PropertyTypeAttribute("ApplicationShortName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")]
public sealed class ApplicationShortName : MessageContextPropertyBase
```

C++

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyGuidAttribute(L"f77a3fcb-4204-41af-8176-a94757e41fe3")]
[PropertyTypeAttribute(L"ApplicationShortName", L"http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", L"string", L"System.String")]
public ref class ApplicationShortName sealed : public MessageContextPropertyBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute IsSensitivePropertyAttribute(false) */
/** @attribute PropertyGuidAttribute("f77a3fcb-4204-41af-8176-a94757e41fe3") */
/** @attribute PropertyTypeAttribute("ApplicationShortName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String") */
public final class ApplicationShortName extends MessageContextPropertyBase
```

JScript

```
SerializableAttribute
IsSensitivePropertyAttribute(false)
PropertyGuidAttribute("f77a3fcb-4204-41af-8176-a94757e41fe3")
PropertyTypeAttribute("ApplicationShortName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")
public final class ApplicationShortName extends MessageContextPropertyBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.PropertyBase](#)

[Microsoft.XLANGs.BaseTypes.MessageContextPropertyBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ApplicationShortName

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationShortName Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationShortName Members

The following tables list the members exposed by the [ApplicationShortName](#) type.

Public Constructors

Name	Description
ApplicationShortName	

[Top](#)

Public Properties

Name	Description
 Name	Overridden.
 QName	(inherited from PropertyBase)
 Type	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[ApplicationShortName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationShortName Constructor

Initializes a new instance of the [ApplicationShortName](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertySchema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public ApplicationShortName ()
```

C++

```
public:  
ApplicationShortName ()
```

J#

```
public ApplicationShortName ()
```

JScript

```
public function ApplicationShortName ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationShortName Class](#)

[ApplicationShortName Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationShortName Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[ApplicationShortName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationShortName Properties

Public Properties

	Name	Description
	Name	Overridden.
	QName	(inherited from PropertyBase)
	Type	Overridden.

See Also

Reference

[ApplicationShortName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ApplicationShortName.Name Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Name As XmlQualifiedName
```

C#

```
public override XmlQualifiedName Name { get; }
```

C++

```
public:  
virtual property XmlQualifiedName^ Name {  
    XmlQualifiedName^ get () override;  
}
```

J#

```
/** @property */  
public XmlQualifiedName get_Name ()
```

JScript

```
public override function get Name () : XmlQualifiedName
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationShortName Class](#)

[ApplicationShortName Members](#)

ApplicationShortName.Type Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Type As Type
```

C#

```
public override Type Type { get; }
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_Type ()
```

JScript

```
public override function get Type () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ApplicationShortName Class](#)

[ApplicationShortName Members](#)

Language Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<IsSensitivePropertyAttribute(False)> _
<PropertyTypeAttribute("Language", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")> _
<PropertyGuidAttribute("8BC62754-297D-4908-9332-4E2A619483DA")> _
Public NotInheritable Class Language
    Inherits MessageContextPropertyBase
```

C#

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyTypeAttribute("Language", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")]
[PropertyGuidAttribute("8BC62754-297D-4908-9332-4E2A619483DA")]
public sealed class Language : MessageContextPropertyBase
```

C++

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyTypeAttribute(L"Language", L"http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", L"string", L"System.String")]
[PropertyGuidAttribute(L"8BC62754-297D-4908-9332-4E2A619483DA")]
public ref class Language sealed : public MessageContextPropertyBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute IsSensitivePropertyAttribute(false) */
/** @attribute PropertyTypeAttribute("Language", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String") */
/** @attribute PropertyGuidAttribute("8BC62754-297D-4908-9332-4E2A619483DA") */
public final class Language extends MessageContextPropertyBase
```

JScript

```
SerializableAttribute
IsSensitivePropertyAttribute(false)
PropertyTypeAttribute("Language", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")
PropertyGuidAttribute("8BC62754-297D-4908-9332-4E2A619483DA")
public final class Language extends MessageContextPropertyBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.PropertyBase](#)

[Microsoft.XLANGs.BaseTypes.MessageContextPropertyBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.Language

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Language Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

Language Members

The following tables list the members exposed by the [Language](#) type.

Public Constructors

Name	Description
Language	

[Top](#)

Public Properties

Name	Description
 Name	Overridden.
 QName	(inherited from PropertyBase)
 Type	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[Language Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

Language Constructor

Initializes a new instance of the [Language](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertySchema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public Language ()
```

C++

```
public:  
Language ()
```

J#

```
public Language ()
```

JScript

```
public function Language ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Language Class](#)

[Language Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

Language Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[Language Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

Language Properties

Public Properties

	Name	Description
	Name	Overridden.
	QName	(inherited from PropertyBase)
	Type	Overridden.

See Also

Reference

[Language Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

Language.Name Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Name As XmlQualifiedName
```

C#

```
public override XmlQualifiedName Name { get; }
```

C++

```
public:  
virtual property XmlQualifiedName^ Name {  
    XmlQualifiedName^ get () override;  
}
```

J#

```
/** @property */  
public XmlQualifiedName get_Name ()
```

JScript

```
public override function get Name () : XmlQualifiedName
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Language Class](#)

[Language Members](#)

Language.Type Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property Type As Type
```

C#

```
public override Type Type { get; }
```

C++

```
public:
virtual property Type^ Type {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_Type ()
```

JScript

```
public override function get Type () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Language Class](#)

[Language Members](#)

OrganizationId Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<IsSensitivePropertyAttribute(False)> _
<PropertyGuidAttribute("35247e23-2b8b-45dd-8dd7-d9e9c0965827")> _
<PropertyTypeAttribute("OrganizationId", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")> _
Public NotInheritable Class OrganizationId
    Inherits MessageContextPropertyBase
```

C#

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyGuidAttribute("35247e23-2b8b-45dd-8dd7-d9e9c0965827")]
[PropertyTypeAttribute("OrganizationId", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")]
public sealed class OrganizationId : MessageContextPropertyBase
```

C++

```
[SerializableAttribute]
[IsSensitivePropertyAttribute(false)]
[PropertyGuidAttribute(L"35247e23-2b8b-45dd-8dd7-d9e9c0965827")]
[PropertyTypeAttribute(L"OrganizationId", L"http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", L"string", L"System.String")]
public ref class OrganizationId sealed : public MessageContextPropertyBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute IsSensitivePropertyAttribute(false) */
/** @attribute PropertyGuidAttribute("35247e23-2b8b-45dd-8dd7-d9e9c0965827") */
/** @attribute PropertyTypeAttribute("OrganizationId", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String") */
public final class OrganizationId extends MessageContextPropertyBase
```

JScript

```
SerializableAttribute
IsSensitivePropertyAttribute(false)
PropertyGuidAttribute("35247e23-2b8b-45dd-8dd7-d9e9c0965827")
PropertyTypeAttribute("OrganizationId", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")
public final class OrganizationId extends MessageContextPropertyBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.PropertyBase](#)

[Microsoft.XLANGs.BaseTypes.MessageContextPropertyBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.OrganizationId

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OrganizationId Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

OrganizationId Members

The following tables list the members exposed by the [OrganizationId](#) type.

Public Constructors

Name	Description
OrganizationId	

[Top](#)

Public Properties

Name	Description
 Name	Overridden.
 QName	(inherited from PropertyBase)
 Type	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[OrganizationId Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

OrganizationId Constructor

Initializes a new instance of the [OrganizationId](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public OrganizationId ()
```

C++

```
public:  
OrganizationId ()
```

J#

```
public OrganizationId ()
```

JScript

```
public function OrganizationId ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OrganizationId Class](#)

[OrganizationId Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

OrganizationId Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[OrganizationId Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

OrganizationId Properties

Public Properties

	Name	Description
	Name	Overridden.
	QName	(inherited from PropertyBase)
	Type	Overridden.

See Also

Reference

[OrganizationId Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

OrganizationId.Name Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Name As XmlQualifiedName
```

C#

```
public override XmlQualifiedName Name { get; }
```

C++

```
public:  
virtual property XmlQualifiedName^ Name {  
    XmlQualifiedName^ get () override;  
}
```

J#

```
/** @property */  
public XmlQualifiedName get_Name ()
```

JScript

```
public override function get Name () : XmlQualifiedName
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OrganizationId Class](#)

[OrganizationId Members](#)

OrganizationId.Type Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
Public Overrides ReadOnly Property Type As Type
```

C#

```
public override Type Type { get; }
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_Type ()
```

JScript

```
public override function get Type () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[OrganizationId Class](#)

[OrganizationId Members](#)

ResponsibilityKey Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<PropertyGuidAttribute("74E011E2-6945-4ce3-87DE-987C453F4F0F")> _
<PropertyTypeAttribute("ResponsibilityKey", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")> _
<IsSensitivePropertyAttribute(False)> _
Public NotInheritable Class ResponsibilityKey
    Inherits MessageContextPropertyBase
```

C#

```
[SerializableAttribute]
[PropertyGuidAttribute("74E011E2-6945-4ce3-87DE-987C453F4F0F")]
[PropertyTypeAttribute("ResponsibilityKey", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")]
[IsSensitivePropertyAttribute(false)]
public sealed class ResponsibilityKey : MessageContextPropertyBase
```

C++

```
[SerializableAttribute]
[PropertyGuidAttribute(L"74E011E2-6945-4ce3-87DE-987C453F4F0F")]
[PropertyTypeAttribute(L"ResponsibilityKey", L"http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", L"string", L"System.String")]
[IsSensitivePropertyAttribute(false)]
public ref class ResponsibilityKey sealed : public MessageContextPropertyBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute PropertyGuidAttribute("74E011E2-6945-4ce3-87DE-987C453F4F0F") */
/** @attribute PropertyTypeAttribute("ResponsibilityKey", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String") */
/** @attribute IsSensitivePropertyAttribute(false) */
public final class ResponsibilityKey extends MessageContextPropertyBase
```

JScript

```
SerializableAttribute
PropertyGuidAttribute("74E011E2-6945-4ce3-87DE-987C453F4F0F")
PropertyTypeAttribute("ResponsibilityKey", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")
IsSensitivePropertyAttribute(false)
public final class ResponsibilityKey extends MessageContextPropertyBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.PropertyBase](#)

[Microsoft.XLANGs.BaseTypes.MessageContextPropertyBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ResponsibilityKey

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityKey Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityKey Members

The following tables list the members exposed by the [ResponsibilityKey](#) type.

Public Constructors

Name	Description
ResponsibilityKey	

[Top](#)

Public Properties

Name	Description
 Name	Overridden.
 QName	(inherited from PropertyBase)
 Type	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[ResponsibilityKey Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityKey Constructor

Initializes a new instance of the [ResponsibilityKey](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public ResponsibilityKey ()
```

C++

```
public:  
ResponsibilityKey ()
```

J#

```
public ResponsibilityKey ()
```

JScript

```
public function ResponsibilityKey ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityKey Class](#)

[ResponsibilityKey Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityKey Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[ResponsibilityKey Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityKey Properties

Public Properties

	Name	Description
	Name	Overridden.
	QName	(inherited from PropertyBase)
	Type	Overridden.

See Also

Reference

[ResponsibilityKey Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityKey.Name Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Name As XmlQualifiedName
```

C#

```
public override XmlQualifiedName Name { get; }
```

C++

```
public:  
virtual property XmlQualifiedName^ Name {  
    XmlQualifiedName^ get () override;  
}
```

J#

```
/** @property */  
public XmlQualifiedName get_Name ()
```

JScript

```
public override function get Name () : XmlQualifiedName
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityKey Class](#)

[ResponsibilityKey Members](#)

ResponsibilityKey.Type Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration
Public Overrides ReadOnly Property Type As Type
```

C#

```
public override Type Type { get; }
```

C++

```
public:
virtual property Type^ Type {
    Type^ get () override;
}
```

J#

```
/** @property */
public Type get_Type ()
```

JScript

```
public override function get Type () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityKey Class](#)

[ResponsibilityKey Members](#)

ResponsibilityName Class

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
<PropertyGuidAttribute("0e8f522c-17ef-4d7b-ae5c-1d406dac9384")> _
<PropertyTypeAttribute("ResponsibilityName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")> _
<IsSensitivePropertyAttribute(False)> _
Public NotInheritable Class ResponsibilityName
    Inherits MessageContextPropertyBase
```

C#

```
[SerializableAttribute]
[PropertyGuidAttribute("0e8f522c-17ef-4d7b-ae5c-1d406dac9384")]
[PropertyTypeAttribute("ResponsibilityName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")]
[IsSensitivePropertyAttribute(false)]
public sealed class ResponsibilityName : MessageContextPropertyBase
```

C++

```
[SerializableAttribute]
[PropertyGuidAttribute(L"0e8f522c-17ef-4d7b-ae5c-1d406dac9384")]
[PropertyTypeAttribute(L"ResponsibilityName", L"http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", L"string", L"System.String")]
[IsSensitivePropertyAttribute(false)]
public ref class ResponsibilityName sealed : public MessageContextPropertyBase
```

J#

```
/** @attribute SerializableAttribute() */
/** @attribute PropertyGuidAttribute("0e8f522c-17ef-4d7b-ae5c-1d406dac9384") */
/** @attribute PropertyTypeAttribute("ResponsibilityName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String") */
/** @attribute IsSensitivePropertyAttribute(false) */
public final class ResponsibilityName extends MessageContextPropertyBase
```

JScript

```
SerializableAttribute
PropertyGuidAttribute("0e8f522c-17ef-4d7b-ae5c-1d406dac9384")
PropertyTypeAttribute("ResponsibilityName", "http://schemas.microsoft.com/OracleEBS/2008/05/OracleEBSContextProperties", "string", "System.String")
IsSensitivePropertyAttribute(false)
public final class ResponsibilityName extends MessageContextPropertyBase
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.XLANGs.BaseTypes.PropertyBase](#)

[Microsoft.XLANGs.BaseTypes.MessageContextPropertyBase](#)

Microsoft.Adapters.OracleEBS.BiztalkPropertySchema.ResponsibilityName

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityName Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityName Members

The following tables list the members exposed by the [ResponsibilityName](#) type.

Public Constructors

Name	Description
ResponsibilityName	

[Top](#)

Public Properties

Name	Description
 Name	Overridden.
 QName	(inherited from PropertyBase)
 Type	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[ResponsibilityName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityName Constructor

Initializes a new instance of the [ResponsibilityName](#) Class.

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertySchema.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public ResponsibilityName ()
```

C++

```
public:  
ResponsibilityName ()
```

J#

```
public ResponsibilityName ()
```

JScript

```
public function ResponsibilityName ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityName Class](#)

[ResponsibilityName Members](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityName Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[ResponsibilityName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityName Properties

Public Properties

	Name	Description
	Name	Overridden.
	QName	(inherited from PropertyBase)
	Type	Overridden.

See Also

Reference

[ResponsibilityName Class](#)

[Microsoft.Adapters.OracleEBS.BiztalkPropertySchema Namespace](#)

ResponsibilityName.Name Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Name As XmlQualifiedName
```

C#

```
public override XmlQualifiedName Name { get; }
```

C++

```
public:  
virtual property XmlQualifiedName^ Name {  
    XmlQualifiedName^ get () override;  
}
```

J#

```
/** @property */  
public XmlQualifiedName get_Name ()
```

JScript

```
public override function get Name () : XmlQualifiedName
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityName Class](#)

[ResponsibilityName Members](#)

ResponsibilityName.Type Property

Namespace: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema

Assembly: Microsoft.Adapters.OracleEBS.BiztalkPropertySchema (in microsoft.adapters.oracleebs.biztalkpropertyschema.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Type As Type
```

C#

```
public override Type Type { get; }
```

C++

```
public:  
virtual property Type^ Type {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_Type ()
```

JScript

```
public override function get Type () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[ResponsibilityName Class](#)

[ResponsibilityName Members](#)

Microsoft BizTalk Adapter for mySAP Business Suite Help

Welcome to Microsoft BizTalk Adapter for mySAP Business Suite Help, which includes a variety of resources that can help you learn to develop, deploy, administer, and use the SAP adapter.

To download a copy of this Help, go to <http://go.microsoft.com/fwlink/?LinkId=196849>.

	Getting Started Information about new features, limitations, prerequisites, and topics for users who are new to the SAP adapter.
	SAP Adapter Tutorials Step-by-step instructions for learning how to use the SAP adapter for specific scenarios.
	Planning and Architecture Architectural overview of the SAP adapter.
	Development Instructions on how to use the adapters with BizTalk Server, the Windows Communication Foundation (WCF) service model, and the WCF channel model.
	Deployment Information about installing the SAP adapter, migrating from previous versions of the adapter, and troubleshooting the installation.
	Troubleshooting Information about enabling tracing and about troubleshooting issues with the BizTalk Adapter Pack and with the SAP adapter, in particular.
	Samples Samples demonstrating how to use the Microsoft BizTalk Adapter for mySAP Business Suite.
	Community Resources Information about how to get assistance with the BizTalk Adapter Pack.

Microsoft BizTalk Adapter for Siebel eBusiness Applications Help

Welcome to Microsoft BizTalk Adapter for Siebel eBusiness Applications Help, which includes a variety of resources that can help you learn to develop, deploy, administer, and use the Siebel adapter.

To download a copy of this Help, go to <http://go.microsoft.com/fwlink/?LinkId=196849>.

	Getting Started Information about new features, limitations, prerequisites, and topics for users who are new to the Siebel adapter.
	Siebel Adapter Tutorials Step-by-step instructions for learning how to use the Siebel adapter for specific scenarios.
	Planning and Architecture Architectural overview of the Siebel adapter.
	Development Instructions on how to use the adapters with BizTalk Server, the Windows Communication Foundation (WCF) service model, and the WCF channel model.
	Deployment Information about installing the Siebel adapter, migrating from previous versions of the adapter, and troubleshooting the installation.
	Troubleshooting Information about enabling tracing and about troubleshooting issues with the BizTalk Adapter Pack and with the Siebel adapter, in particular.
	Samples Samples demonstrating how to use the Microsoft BizTalk Adapter for Siebel eBusiness Applications.
	Community Resources Information about how to get assistance with the BizTalk Adapter Pack.

Getting Started

This section provides an overview of the adapter, prerequisites, and topics for users who are new to the Microsoft BizTalk Adapter Pack. It discusses the features of Microsoft BizTalk Adapter for Siebel eBusiness Applications and the different operations that can be performed on the Siebel system using the adapter.

What is an adapter? An adapter is a software component that enables you to send and receive messages to and from a line-of-business (LOB) system. The primary design goal of adapters is to facilitate the exchange of business documents between trading partners. Because each business system may adhere to specific document formats and protocols, the adapter must use a delivery mechanism that conforms to commonly recognized standards and protocols.

The adapters can be divided into two broad categories:

- **LOB adapters.** Such adapters provide a service-oriented programming model to access LOB systems—for example, adapters for SAP or Siebel.
- **Data adapters.** Such adapters provide a service-oriented programming model to access databases—for example, an adapter for the Oracle database or SQL Server.

There are five adapters in the BizTalk Adapter Pack:

- Microsoft BizTalk Adapter for SQL Server (the SQL adapter)
- Microsoft BizTalk Adapter for Oracle E-Business Suite (the Oracle E-Business adapter)
- Microsoft BizTalk Adapter for Oracle Database (the Oracle Database adapter)
- Microsoft BizTalk Adapter for mySAP Business Suite (the SAP adapter), including .NET Framework Data Provider for mySAP Business Suite (Data Provider for SAP)
- Microsoft BizTalk Adapter for Siebel eBusiness Applications (the Siebel adapter), including .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel)

Note

The Siebel adapter is not available for 64-bit platforms.

If you do not already know how you want to use the Siebel adapter at your company, it is recommended that you start by exploring the features and functionality of the adapter described in [Overview of BizTalk Adapter for Siebel eBusiness Applications](#).

In This Section

- [Prerequisite Skills and Knowledge](#)
- [Common Administrative Tasks](#)
- [Common Developer Tasks](#)
- [Understanding BizTalk Adapter for Siebel eBusiness Applications](#)
- [Siebel Adapter Tutorials](#)
- [Community Resources](#)
- [Frequently Asked Questions](#)

Prerequisite Skills and Knowledge

The potential users for the BizTalk Adapter Pack are:

- Developers who directly program to the adapters.
- IT professionals who consume the adapters using other integration platforms like Microsoft BizTalk Server and other tools.
- Independent software vendors (ISVs) who build solutions on top of the adapters.

The prerequisite skills and knowledge required for each role are detailed below.

Prerequisites for Developers

A developer using Microsoft BizTalk Adapter for Siebel eBusiness Applications must be at least moderately experienced with the use of:

- Microsoft Visual Studio 2010, with the development of .NET solutions
- Programming with the .NET Framework 3.5 SP1 and .NET Framework 4
- Programming with the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
- Extensible Markup Language (XML)
- XML Schema definition (XSD) language
- Web Services Description Language (WSDL)

Prerequisites for IT Professionals

An IT professional using the Siebel adapter must be at least moderately experienced with the use of:

- SQL Server Integration Services (SSIS)
- Microsoft BizTalk Server 2010
- Windows SharePoint Services

Prerequisites for ISVs

ISVs using the Siebel adapter must be at least moderately experienced with:

- The internal workings and concepts of Siebel to be able to build applications on top of the adapters
- The .NET Framework 3.5 SP1 and .NET Framework 4
- The WCF LOB Adapter SDK

See Also

Other Resources

[Getting Started](#)

[Understanding BizTalk Adapter for Siebel eBusiness Applications](#)

Common Administrative Tasks

The following table provides links to topics helpful for deploying, using, and maintaining Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Administrative Tasks <ul style="list-style-type: none">• Siebel Adapter Tutorials• Deployment	Planning and Architecture <ul style="list-style-type: none">• Architecture for BizTalk Adapter for Siebel eBusiness Applications• Understanding BizTalk Adapter for Siebel eBusiness Applications• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Glossary
---	---

See Also

Concepts

[Common Developer Tasks](#)

Other Resources

[Getting Started](#)

Common Developer Tasks

The following table provides links to topics useful for developing applications on top of Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Developing <ul style="list-style-type: none">• Establishing a Connection to the Siebel System• Developing BizTalk Applications• Developing Applications by Using the WCF Service Model• Developing Applications by Using the WCF Channel Model• Using the .NET Framework Data Provider for Siebel eBusiness Applications	Planning and Architecture <ul style="list-style-type: none">• Architecture for BizTalk Adapter for Siebel eBusiness Applications• Understanding BizTalk Adapter for Siebel eBusiness Applications• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Troubleshooting• Technical Reference• Glossary
---	---

See Also

Concepts

[Common Administrative Tasks](#)

Other Resources

[Getting Started](#)

Understanding BizTalk Adapter for Siebel eBusiness Applications

The BizTalk Adapter Pack enables service-oriented programmatic access in order to interact with an external system. The adapters provide the following advantages to clients:

- **Consistent design-time experience.** The adapters provide a common and user-friendly design time experience for browsing, searching, and retrieving metadata of LOB artifacts.
- **Varied programming options.** The adapters provide a choice of programming model including Windows Communication Foundation (WCF) channel model, WCF service model, ADO.NET, Web services, or BizTalk supported models.
- **Uniform experience across LOBs.** The adapters standardize on using the WCF and Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and hence provide a uniform experience of gaining access to any LOB system.

As mentioned, the adapters are built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK. The WCF LOB Adapter SDK provides a common basis for building integration adapters that a variety of client applications such as BizTalk Server and Microsoft Office can consume. The WCF LOB Adapter SDK aligns the adapter strategy with the Microsoft Services strategy by exposing integration adapters as Windows Communication Foundation (WCF) channels. For more information about the WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation. The documentation is installed along with the WCF LOB Adapter SDK, typically under <installation drive>:\Program Files\WCF LOB Adapter SDK\Documents.

To perform operations on a Siebel system, adapter clients must have access to business services exposed by the Siebel system. A Siebel application exposes data as business components and business objects. A Siebel *business component* is a logical entity that associates columns from one or more tables into a single structure. A Siebel *business object* implements a business model by tying together a set of interrelated business components. With the Microsoft BizTalk Adapter for Siebel eBusiness Applications, adapter clients can surface Siebel business objects and business components.

The Siebel adapter also includes the .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel). The Data Provider for Siebel provides an ADO interface to a Siebel system by extending ADO.NET interfaces.

This section discusses the features of the Siebel adapter and the Data Provider for Siebel.

In This Section

- [Overview of BizTalk Adapter for Siebel eBusiness Applications](#)
- [New Features in the Siebel Adapter](#)
- [Limitations of BizTalk Adapter for Siebel eBusiness Applications](#)
- [About the Data Provider for Siebel](#)

Overview of BizTalk Adapter for Siebel eBusiness Applications

The Microsoft BizTalk Adapter for Siebel eBusiness Applications exposes the Siebel system as a WCF service. Adapter clients can perform operations on the Siebel system by exchanging SOAP messages with the adapter. The adapter consumes the WCF message and makes appropriate calls to the Siebel system to perform the operation. The adapter returns the response from the Siebel system back to the client in the form of SOAP messages.

The Siebel adapter surfaces metadata of Siebel artifacts (business components, business services) that describes the structure of a SOAP message in the form of WSDL. The Siebel adapter uses the Add Adapter Service Reference Visual Studio Plug-in and Consume Adapter Service BizTalk Project Add-in to enable adapter clients to retrieve metadata for operations and generates programming artifacts that can be used in your programming solution.

The Siebel adapter uses the Siebel COM Data Control to access the Siebel system. The Siebel COM Data Control comes bundled with the Siebel Web client. Hence, make sure you have the Siebel Web client installed on the same computer as the Siebel adapter. You can use the Siebel adapter to communicate with the Siebel system in the following ways:

- By developing BizTalk applications. See [Developing BizTalk Applications](#).
- By using the WCF service model. See [Developing Applications by Using the WCF Service Model](#).
- By using the WCF channel model. See [Developing Applications by Using the WCF Channel Model](#).

In This Section

- [How Does the Adapter Connect to a Siebel System?](#)
- [How Does the Adapter Surface Siebel Metadata?](#)
- [What Operations Can be Performed Using the Adapter?](#)
- [Other Features Supported by the Adapter](#)

See Also

Other Resources

[Understanding BizTalk Adapter for Siebel eBusiness Applications](#)

How Does the Adapter Connect to a Siebel System?

The Siebel adapter requires adapter clients to provide a connection string, called the connection Uniform Resource Identifier (URI), to connect to a Siebel system. With a connection URI, adapter clients can specify connection parameters to connect to an external system. For more information about the connection URI, see [Establishing a Connection to the Siebel System](#).

Make sure you comply with the security guidelines when establishing a connection with the Siebel system. For more information about security guidelines, see [Security and Protection](#).

See Also

Concepts

[Overview of BizTalk Adapter for Siebel eBusiness Applications](#)

How Does the Adapter Surface Siebel Metadata?

The Siebel adapter surfaces metadata from the Siebel system that describes the message structure for communicating with a Siebel system using the adapter. The Siebel adapter supports two interfaces for retrieving metadata.

- MetadataExchange provided by Windows Communication Foundation (WCF). WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from the Siebel system.
- IMetadataRetrievalContract provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

The goal of the Microsoft BizTalk Adapter for Siebel eBusiness Applications is to expose the Siebel system as a WCF service. The adapter surfaces Siebel artifacts (business objects and business services) as operations that adapter clients can invoke.

Adapter clients can browse, search, and retrieve metadata by using the WCF service model, by using the WCF channel model, or by creating a BizTalk project in Visual Studio. When using the WCF service model, you must use the Add Adapter Service Reference Visual Studio Plug-in to generate the proxy classes for performing operations on the Siebel system. When using a BizTalk project, you must use the Consume Adapter Service BizTalk Project Add-in to generate metadata for the operations that you want to perform on the Siebel system. For more information about browsing, searching, and retrieving metadata using the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).

Browsing Metadata

The Siebel adapter enables adapter clients to browse the business objects and business services exposed by the Siebel system. As part of the metadata browse operation, the adapter categorizes the artifacts into multiple logical levels and exposes a hierarchical view of metadata to the adapter clients. Metadata from a Siebel system is categorized under the following nodes:

- **Business Objects.** This node contains the business objects in a Siebel system, which is a logical collection of business components. The business objects are further categorized as business components. At the lowest level of the hierarchy are the operations that you can invoke on the business components.
- **Business Services.** This node contains the business services exposed by a Siebel system. A Siebel business service is a collection of business service methods or functions that can be directly invoked in the Siebel system.

For more information about how the metadata is categorized, see [Metadata Node IDs](#).

Searching Metadata

The Siebel adapter enables adapter clients to search metadata in a Siebel system. It does this by using valid Siebel search expressions compatible with Siebel's LIKE operator on the [Name] field of the Siebel repository business component used for metadata browsing. The following table lists the Siebel artifacts and the metadata hierarchy under which they can search.

Artifact	Searches under the node in the GUI
Business Object	/Business Objects
Business Component	/Business Objects/Business Object Name
Business Service	/Business Services

The following table lists the special characters that can be used for search and their interpretation by the Siebel adapter.

Special character	Interpretation
? (question mark)	Matches exactly one character. For example, A? matches AB, AC, AD

* (asterisk)	Matches zero or more characters. For example, A* matches A, AB, ABC.
--------------	---

Retrieving Metadata

When retrieving metadata, the Siebel adapter can extract metadata under a schema, including all or a subset of business objects or business services with the respective operation parameters. It presents the entities from the Siebel system as element names in XML. The Siebel adapter enables adapter clients to retrieve metadata for the Siebel system, including detailed metadata characteristics:

- For business components, the Siebel adapter retrieves such items as the business object name, business component name, the field names, data types, field lengths, mandatory field, optional field, and picklist field. The adapter also retrieves the operations possible on the business component such as INSERT, UPDATE, DELETE, and QUERY.
- For business service methods, the Siebel adapter retrieves such items as the business service name, the method name (same as operation), method parameters, and parameter data types.

For more information about retrieving the metadata, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).

See Also

Concepts

[Overview of BizTalk Adapter for Siebel eBusiness Applications](#)

What Operations Can be Performed Using the Adapter?

Adapter clients can perform operations on the Siebel system by either:

- Creating BizTalk projects.
- Using the WCF channel model.
- Using the WCF service model.

The Siebel adapter exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel. For information about the message structure and the SOAP action associated with each operation, see [Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#).

This section provides information about the operations supported on the Siebel system using the Siebel adapter.

Note

The Siebel adapter does not support receiving inbound calls from a Siebel system.

In This Section

- [Operations on Business Components](#)
- [Operations on Business Services](#)

See Also

Concepts

[Overview of BizTalk Adapter for Siebel eBusiness Applications](#)

Operations on Business Components

A Siebel business component is a logical entity that associates columns from one or more database tables into a single structure. Adapter clients can perform the following operations on the Siebel business components by using the adapter:

- **Insert.** Adapter clients can insert one or more records into a business component.
- **Query.** Adapter clients can query one or more records from a business component. This operation takes the following parameters as input:
 - SearchExpr: Contains a search expression. All records under a specified business component are compared against this search expression, and matching records are returned to the adapter client.
 - SortSpec: If there are multiple records that match the search expression, this specification determines the order in which records are returned. This parameter is optional.
 - QueryFields: Enables adapter clients to retrieve values for only a subset of fields in returned records. This parameter is optional.

Note

The Siebel adapter supports using original names in the QueryField parameter in the Query operation on the business component

- **Update.** Adapter clients can update one or more records in a business component.
- **Delete.** Adapter clients can delete one or more records in a business component by specifying a set of IDs or by providing a search expression.

Note

In addition to other parameters the Query, Update, and Delete operations also take a ViewMode parameter. This parameter takes an integer that determines the access permissions of the user. For more information about the ViewMode parameter and the other parameters for these operations, see the request message for business component operations under [Message Schemas for Business Component Operations](#).

For information about:

- Performing operations on business components using BizTalk Server, see [Performing Operations on Business Components Using BizTalk Server](#).
- Performing operations on business components using the WCF service model, see [Performing Operations on Business Components by Using the WCF Service Model](#).
- Performing Operations on business components using the WCF channel model, see [Performing Operations on Business Components by Using the WCF Channel Model](#).
- Performing operations on business components using message structures and SOAP actions, see [Message Schemas for Business Component Operations](#).

Operations on Business Components with MVG Fields

A Siebel business component can also retrieve fields from other business components using joins or multivalued groups (MVG). In addition to the Insert, Query, Update, and Delete operations that are surfaced for all business components, adapter clients can perform the following operations on the Siebel business components by using the adapter:

- **Associate.** Adapter clients can associate records by specifying search expressions for parent and child records. This is applicable only for business components with MVG fields. The search expressions should filter exactly one record for both the parent and child business components.
- **Disassociate.** Adapter clients can dissociate records by specifying search expressions for parent and child records. This is applicable only for business components with MVG fields. The search expressions must filter exactly one record for both the parent and child business components.
- **Query_[MVG_Child_Business_Comp].** Adapter clients can query the child records that are associated with a parent record by specifying the parent record and the MVG field name. This is applicable only for business components with MVG fields.

 **Note**

In addition to other parameters, these operations also take a ViewMode parameter. This parameter takes an integer that determines the access permissions of the user. For more information about the ViewMode parameter and the other parameters for these operations, see the request message for business component operations under [Message Schemas for Business Component Operations](#).

For more information about:

- Performing these operations on business components using BizTalk Server, see [Performing Operations on Business Components with MVG Fields Using BizTalk Server](#).
- Performing these operations on business components using WCF service model, see [Performing Operations on Business Components with MVG Fields by Using the WCF Service Model](#).
- Message structures and SOAP actions for these operations, see [Message Schemas for Business Component Operations](#).

Operations on Business Components with Picklist Fields

Picklist field types in business components constitute a collection of values from which users can pick specific values to pass to the Siebel system. The Siebel adapter supports operations on a business component with picklist fields. The operations on business components containing picklist fields are the same as operations on any other business component, as described in the preceding paragraph. However, depending on the kind of picklist, the input values to the business component may vary. For more information about picklists and their types, refer to Siebel documentation.

One of the picklist types, static bounded picklists, are surfaced by the adapters as an enumerated picklist type in the metadata generated by the adapter at design time. This contains a static set of values that must be specified for the picklist type at run time. While specifying a value for a static bounded picklist, you must always specify a value that belongs to the set.

The message structure to perform operations on business components with picklist fields is similar to the message structures for operations on any other business component, as described in [Message Schemas for Business Component Operations](#).

For more information about:

- Message structures for business components containing picklist fields, see [Message Schema for Picklist Operations](#).
- Performing operations on a business component that contains picklist fields, see [Performing Operations on Business Components with Picklist Fields Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Operations on Business Services

A Siebel business service is a collection of business methods that can be directly invoked in Siebel. Whereas business components and business objects are associated to specific data and tables in Siebel, business services invoke the objects to perform specific tasks.

The Siebel adapter surfaces the business service methods as operation names and supports IN, OUT, and INOUT parameters. For example, the Siebel adapter surfaces the **ATPRunCheck** method as an operation. This method belongs to the **ATP** business service.

Certain conditions that the Siebel adapter imposes while using the Siebel business services:

- If a Siebel business service method takes an argument that does not have the data type specified, the adapter exposes the argument as TEXT.
- A Siebel business service method argument can be of the following types:
 - String (exposed as xsd:string)
 - Number (exposed as xsd:decimal)
 - Date (exposed as xsd:DateTime, with pattern facet indicating that time part must be set to 00:00:00)
 - Hierarchy (exposed as xsd:string)
 - Integration Object (exposed as xsd:string)

Also, the business service method verifies whether the value passed for an argument complies with the corresponding type. So, if a business service method accepts or returns values that do not comply with the corresponding argument type, the adapter may throw an exception. If the adapter does succeed in invoking the business service method, the schema validation may fail.

For information about:

- Performing operations on business services using BizTalk Server, see [Invoking Business Service Methods Using BizTalk Server](#).
- Message structures and SOAP actions to perform operations on business services, see [Message Schemas for Business Service Operations](#).

See Also

Other Resources

[What Operations Can be Performed Using the Adapter?](#)

Other Features Supported by the Adapter

In addition to the features discussed throughout the topics of [Overview of BizTalk Adapter for Siebel eBusiness Applications](#), the Siebel adapter provides the following features that are useful for adapter clients:

- **Support for configuring adapters using binding properties.** Adapter clients can configure the Siebel adapter by specifying certain binding properties while generating the metadata. For more information, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
- **Support for null values for operation parameters.** The Siebel adapter enables adapter clients to provide null values for business object operation parameters using the XSD "nillable" attribute. The adapter does not pass fields with null values to the Siebel system.
- **Support for XML data streaming.** Adapter clients can stream data from or to the Siebel adapter by using the **XMLReader** or **XMLWriter** interfaces.
- **Support for dynamic ports in BizTalk Server.** Through the BizTalk WCF-Custom adapter, the Siebel adapter supports a dynamic port that enables dynamic routing of messages from BizTalk Server based on the message context properties. For more information, see [Configuring Dynamic Ports](#).
- **Support for message versioning.** The Siebel adapter supports message versioning. This enables support for different message schemas in future releases of the BizTalk Adapter Pack. For more information, see [Message Versioning Support](#).
- **Support for performance counters.** The Siebel adapter supports WCF-based performance counters that adapter clients can use. For more information about performance counters, see [Using Performance Counters](#).

Note

This feature does not provide backward compatibility with the earlier versions of the adapter.

- **Support for encoding XML element names.** The Siebel adapter presents entities from enterprise applications as element names in XML. Underscores are the only special characters that can be included in the element names. Therefore, underscores are used to encode element names with special characters. For example, `emp$name` is encoded to `emp_x0024_name`.

See Also

Concepts

[Overview of BizTalk Adapter for Siebel eBusiness Applications](#)

New Features in the Siebel Adapter

This section lists the new features in Microsoft BizTalk Adapter for Siebel eBusiness Applications.

New Features in the Siebel Adapter

The following are the new features introduced in this release of the Siebel adapter.

Technology-Related Features

Feature	Comment
Support for using the Siebel adapter with Microsoft Office SharePoint Server (MOSS)	You can use the adapters to present data from the Siebel system onto a MOSS portal. For more information, see Using the Siebel Adapter with Microsoft Office SharePoint Server .

Operations-Related Features

Feature	Comment
Support for ASSOCIATE operation on business components	Adapter clients can associate records by specifying search expressions for parent and child records. This is applicable only for business components with multivalued group (MVG) fields. Note that the search expressions should filter exactly one record for both the parent and child business components.
Support for DISSOCIATE operation on business components	Adapter clients can dissociate records by specifying search expressions for parent and child records. This is applicable only for business components with multivalued group (MVG) fields. Note that the search expressions must filter exactly one record for both the parent and child business components.
Support for multivalued link queries	Adapter clients can query the child records associated with a parent record by specifying the parent record and the multivalued field name. This is applicable only for business components with multivalued group (MVG) fields.

Other Features

Feature	Comment
New way of using the adapter in BizTalk Server	The Siebel adapter can be used in BizTalk either as a WCF-Custom port or a WCF-Siebel port. If you want to use the Siebel adapter through a WCF-Custom port, you do not need to add the WCF-Custom port to the BizTalk Server Administration console because the WCF-Custom port is added to the BizTalk Server Administration console by default. However, if you want to use the Siebel adapter through a WCF-Siebel port, you must first add the WCF-Siebel adapter to the BizTalk Server Administration console. For more information, see Adding the Siebel Adapter to BizTalk Server Administration Console .

See Also

Other Resources

[Understanding BizTalk Adapter for Siebel eBusiness Applications](#)

Limitations of BizTalk Adapter for Siebel eBusiness Applications

The following are known limitations of the Microsoft BizTalk Adapter for Siebel eBusiness Applications:

- The Siebel adapter is not compatible with Microsoft BizTalk Adapter 2.0 for Siebel eBusiness Applications, the previous release of the adapter. The current release of the adapter does not support sending and receiving messages that have schemas generated by using the earlier version of the adapter.

Note

You can modify the BizTalk projects for previous version of the Siebel adapter to use the new WCF-based Siebel adapter. For more information, see [Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#).

- The Siebel adapter does not support workflow objects.
- The Siebel adapter does not validate the format in which a client passes a time value to the Siebel system. Adapter clients must make sure that the value specified for a date and time field adheres to a format in which the Siebel system expects.
- The Siebel adapter does not perform schema validation. For example, a field of length 30 can take values with length 100, if allowed by the Siebel system. It can also lead to a loss of data in certain scenarios because the data that the client inserts through business objects might not necessarily be the data that is actually written to the database. Adapter clients must explicitly validate the input against the schema that is surfaced by the adapter. However, the Siebel adapter does validate that all required fields (for business components) or arguments (for business services) are specified.
- The Siebel adapter expects the time values to be specified in standard Siebel format—that is HH:MM:SS. Time values that are specified in any other format will produce an error, and the Siebel adapter throws a `TargetSystemException`.
- In certain scenarios, the Siebel application might or might not throw an error message. For example, a search operation using an expression might either throw an exception or return zero records. Accordingly, the Siebel adapter might throw a `TargetSystemException` or get an empty XML as the output.
- While retrieving data from the Siebel system using the WCF service model, the Siebel adapter does not deserialize XMLs having more than 65536 nodes. Make sure the output XML has nodes less than or equal to 65536. You can work around this limitation by modifying the app.config file for your application. For instructions, see [Troubleshooting Operational Issues](#).
- The Siebel adapter retrieves the maximum length for a field from the business component layer, as opposed to the database layer. So, if you try to insert a value that complies with the maximum length of the database column, but which is greater than the maximum length of the corresponding field for a business component, the value written to the database might be different from the value you wanted to enter.
- While performing batch operations (Insert, Update, and Delete), the Siebel adapter throws an error if the first operation results in an error. However, if the first operation succeeds and any of the subsequent operations fail, the adapter does not throw an error but rather returns the IDs for records that correspond to the successful operations in the output. Adapter clients must explicitly verify whether all the operations have been successful.
- Due to issues with timeout handling by the underlying Siebel client API, the Siebel adapter does not support command and connection timeout.
- Consider a scenario where user "A" generates the metadata for an operation in Siebel. Another user "B", having lesser privileges than user "A," will be able to access the metadata. The Siebel adapter does not perform any checks to validate whether user "B" must get access to the metadata. However, because of insufficient privileges, user "B" might not be able to execute any operation on the Siebel system by using the metadata.

- The Siebel adapter does not support specifying a connection URI that has special characters for any of the parameter values. For each parameter value that contains special characters, make sure you replace the special characters with the corresponding values, as specified by the URI encoding standards.
- When using the adapters with BizTalk Server, if the credentials on the WCF-Custom send port are incorrect, the request messages are not processed. After you specify the correct credentials, the message is sent to the Siebel system, and a response is received. However, the response message is not available to the out port. In such scenarios, you might need to restart the host instance.

See Also

Other Resources

[Understanding BizTalk Adapter for Siebel eBusiness Applications](#)

About the Data Provider for Siebel

This section lists the various uses of the .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel). For instructions about how to use the Data Provider for Siebel, see [Using the .NET Framework Data Provider for Siebel eBusiness Applications](#).

The Data Provider for Siebel is built on top of the Microsoft BizTalk Adapter for Siebel eBusiness Applications. You can use the Data Provider for Siebel in the following ways:

- To write an ADO.NET client to connect to the Siebel system. The Data Provider for Siebel exposes certain classes that enable you to interface with the provider.
- To perform a SELECT query on a Siebel business component.
- To perform an EXEC query on a Siebel business service.
- To use the Data Provider for Siebel with SQL Server Integration Services (SSIS).

See [Using the .NET Framework Data Provider for Siebel eBusiness Applications](#) for information about:

- The ADO.NET interfaces extended by the Data Provider for Siebel
- The connection string to connect to a Siebel system
- Syntax for the SELECT and EXEC statements
- Using the Data Provider for Siebel with SSIS

In This Section

- [Limitations of the Data Provider for Siebel](#)

Limitations of the Data Provider for Siebel

The following are known limitations of the .NET Framework Data Provider for Siebel eBusiness Applications:

- The Data Provider for Siebel supports alias names for tables in the SELECT clause, but not in the WHERE clause.
- The Data Provider for Siebel fails to create a table with column names that have the special character, "]". You can escape the special character by including another closing square bracket. So, you should include"]]" instead of "]".
- Due to issues with timeout handling by the underlying Siebel client API, the Data Provider for Siebel does not support command and connection timeout.
- The Data Provider for Siebel does not support asynchronous command behavior.
- When used with a SQL Server Integration Services (SSIS) project, the Data Provider for Siebel fails to retrieve data for columns that contain values with more than 8000 characters. This is due to an SSIS restriction according to which:
 - Values greater than 4000 characters in SSIS variable are not supported.
 - Values greater than 4000 wide characters are not supported.
 - Values greater than 8000 single-byte characters are not supported.
- The EXEC operation will not be functional while using the .NET Framework Data Provider for Siebel eBusiness Applications with SQL Server Integration Services (SSIS). So, for example, adapter clients will not be able to execute a business service in Siebel (using Data Provider for Siebel) while using the data providers with SSIS.

See Also

Other Resources

[About the Data Provider for Siebel](#)

Siebel Adapter Tutorials

The first of the Siebel adapter tutorials contains a step-by-step demonstration of how to present data from a Siebel system on a Microsoft SharePoint portal. The second Siebel adapter tutorial contains detailed instructions on how to migrate BizTalk projects created using the previous version of the Siebel adapter to use the WCF-based Siebel adapter.

Also, the topics in the [Developing BizTalk Applications](#) section provide detailed instructions on how to use the Siebel adapter with BizTalk Server.

In This Section

- [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#)
- [Tutorial 2: Migrating BizTalk Projects](#)

See Also

Other Resources

[Getting Started](#)

Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site

This tutorial provides detailed instructions on using the Siebel adapter with Microsoft Office SharePoint Server to present business data from a Siebel system on a SharePoint portal. To demonstrate how to use the Siebel adapter with Office SharePoint Server, we create an application in Office SharePoint Server to retrieve a list of customer accounts from the Siebel repository using the Siebel adapter.

To extract this information from the Siebel system, the example invokes the Query method on the **Account** business components.

Note

Before proceeding with the tutorial, make sure you have installed all the prerequisites for using the Siebel adapter with Office SharePoint Server. For more information about the prerequisites, see the BizTalk Adapter Pack installation guide, typically installed at <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

In This Section

- [Step 1: Publish the Siebel Business Component Operations as a WCF Service](#)
- [Step 2: Create an Application Definition File for Siebel Business Component Operations](#)
- [Step 3: Create a SharePoint Application to Retrieve Data from Siebel](#)
- [Step 4: Test Your SharePoint Application](#)

See Also

Other Resources

[Siebel Adapter Tutorials](#)

Step 1: Publish the Siebel Business Component Operations as a WCF Service



Step 1

Time to complete: 10 minutes

Objective: You can use the WCF Adapter Service Development Wizard to generate a WCF service that can be hosted in a hosting environment such as Internet Information Services (IIS) or Windows Process Activation Service (WAS). This topic demonstrates how to use the wizard to generate a WCF service file.

Prerequisites

Before running the wizard, install the following:

- Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK either with the **Complete** option or the **Custom** option (and choosing **Tools** within this option). This installs the Visual Studio template for the WCF Adapter Service Development Wizard.
- Microsoft BizTalk Adapter for Siebel eBusiness Applications
- The required Siebel client.

For more information about these prerequisites, see the BizTalk Adapter Pack installation guide. The installation guide is typically installed at <installation drive>\Program Files\Microsoft BizTalk Adapter Pack\Documents.

To publish the Siebel Business Components as a WCF Service

1. Start Visual Studio, and then create a project.
2. In the **New Project** dialog box, from the **Project types** pane, select **Visual C#**. From the **Templates** pane, select **WCF Adapter Service**.

Alternatively, from the **Project types** pane, expand **Visual C#**, and then select **Web**. From the **Templates** pane, select **WCF Adapter Service**.

 **Note**

If you installed Visual Studio 2010 with the Web Development component, the **WCF Adapter Service** template is also available from the **New Website** option.

3. Specify a name and location for the solution, and then click **OK**. The WCF Adapter Service Development Wizard starts.
4. On the Welcome page, click **Next**.
5. On the Choose Operations page, specify a connection string to connect to the Siebel system. To do so:
 - a. In the **Select a binding** list, click **siebelBinding**, and then click **Configure**.
 - b. In the **Configure Adapter** dialog box, click the **Security** tab.
 - c. In the **Client credential type** list, select **Username**, and then specify the user name and password to connect to the Siebel system.
 - d. Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).

 **Note**

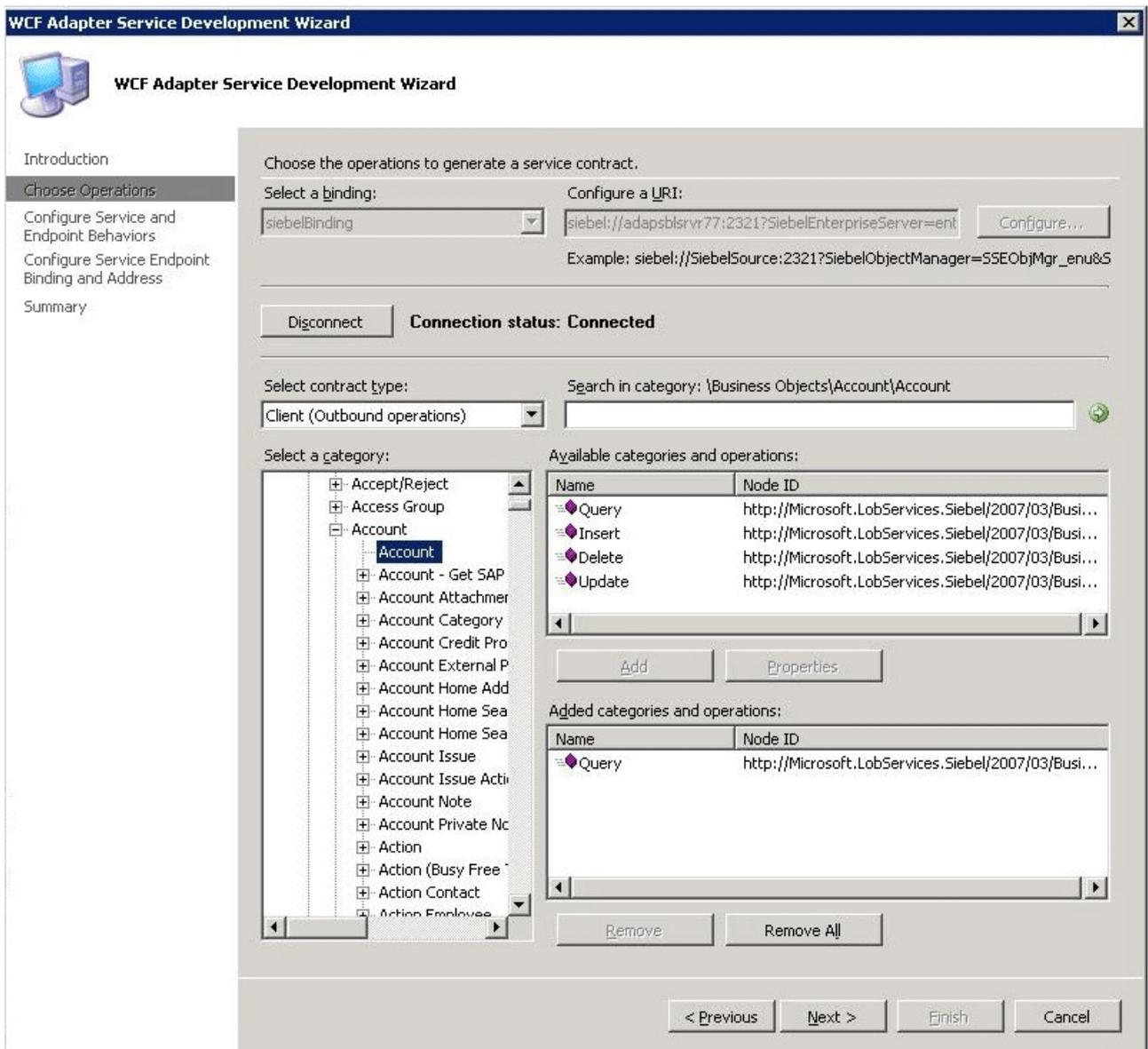
If the connection parameters contain any reserved characters (such as XML special characters), you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

- e. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required for the operations you want to target.

For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

- f. Click **OK**, and then click **Connect**. After the connection is established, the connection status is shown as **Connected**.

- 6. On the Choose Operations page, in the **Select contract type** list, click **Client (Outbound operations)**.
- 7. In the **Select a category** box, expand the Siebel **Business Objects** node to see the list of business objects in the Siebel repository. For this example, do the following:
 - a. Expand the **Account** business object, and then click the **Account** business component.
 - b. In the **Available categories and operations** box, select the **Query** operation, and then click **Add**. The selected operation is listed in the **Added categories and operations** box.



- 8. On the Choose Operations page, click **Next**

8. On the Choose Operations page, click **Next**.

9. On the Configure Service and Endpoint Behaviors page, specify values to configure the service and endpoint behavior.

a. In the **Service Behavior Configuration** box, specify values for the following:

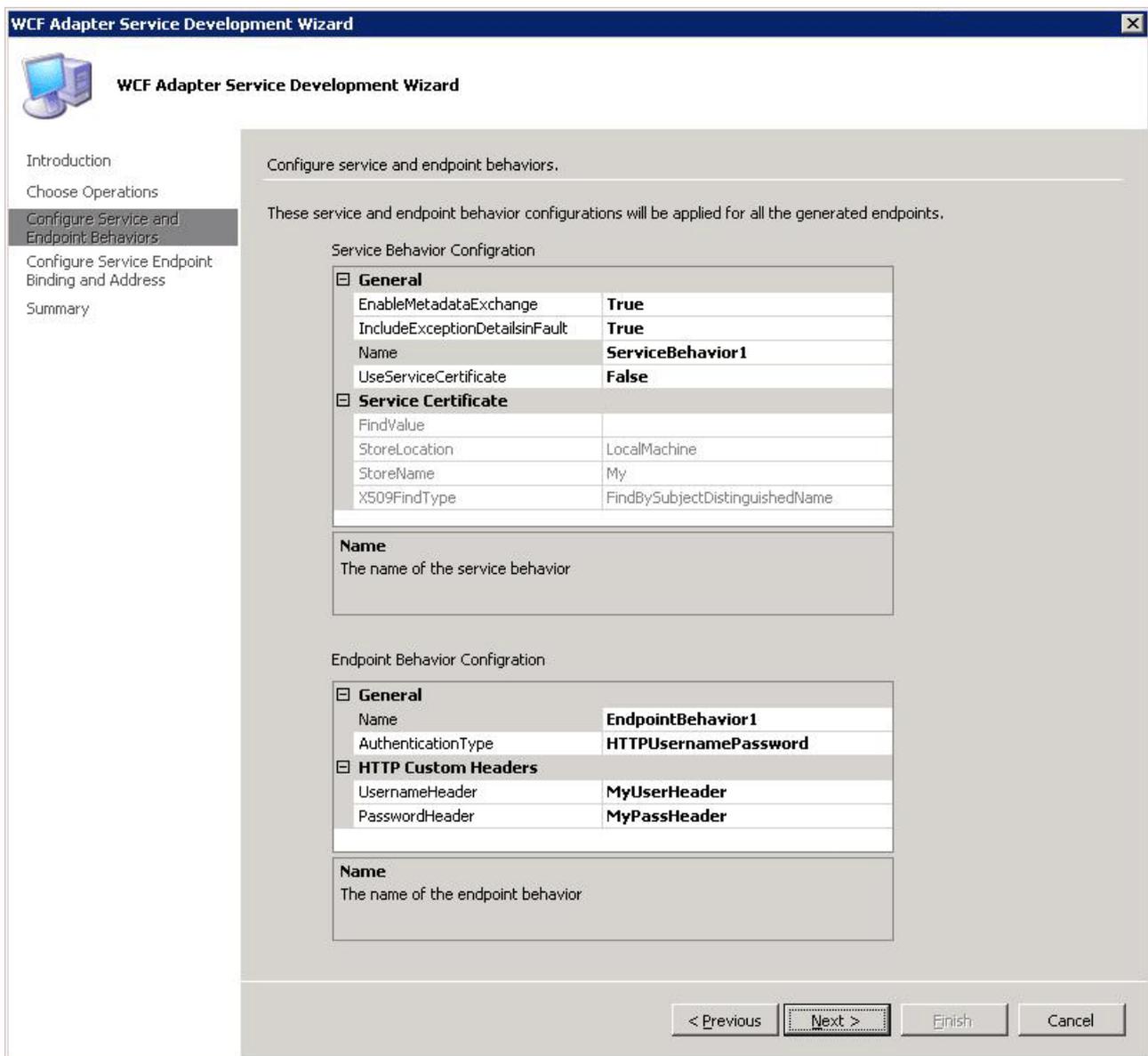
For the property	Specify the value
EnableMetadataExchange	Set this to True to create a metadata exchange endpoint. By setting this to True , you make the service metadata available using standardized protocols, such as WS-Metadata Exchange (MEX) and HTTP/GET requests. Default is False .
IncludeExceptionDetailsInFault	Set this to True to include managed exception information in detail of SOAP faults returned to the client for debugging purposes. Default is False .
Name	Name for the service behavior configuration.
UseServiceCertificate	Specifies whether you want to use the message level security mode of WCF. Default is True . For this tutorial, you must set this to False .
FindValue	A string that specifies the value to search for in the X.509 certificate store.  Note Specify a value for this property only if UseServiceCertificate is set to True .
StoreLocation	A value that specifies the location of the certificate store that the service can use to validate the client's certificate.  Note Specify a value for this property only if UseServiceCertificate is set to True .
StoreName	Name of the X.509 certificate store to open.  Note Specify a value for this property only if UseServiceCertificate is set to True .
X509FindType	The type of X.509 search to be executed.  Note Specify a value for this property only if UseServiceCertificate is set to True .
 Note	
For more information about the certificates and the associated properties, see "X509ClientCertificateCredentialsElement Properties" at http://go.microsoft.com/fwlink/?LinkId=103771 .	

b. In the **Endpoint Behavior Configuration** box, specify values for the following:

For the property	Specify the value

Authentication Type	<ul style="list-style-type: none"> • Set this to ClientCredentialUserNamePassword to enable the clients to specify the user name and password while consuming the WCF service. • Set this to HTTPUserNamePassword to enable clients to specify user name and password as part of the HTTP header. • Set this to Auto to first enable clients to specify credentials through the ClientCredential interface. If this fails, clients can pass credentials as part of the HTTP header. <p>Default is Auto. For Microsoft Office SharePoint Server to consume the WCF service, you should set this as HTTPUserNamePassword.</p>
Name	Specify a name for the endpoint behavior configuration.
UsernameHeader	<p>Name for the user name header. For this example, specify MyUserHeader. For more information about HTTP headers, see "Support for Custom HTTP and SOAP Headers" at http://go.microsoft.com/fwlink/?LinkId=106692.</p> <p> Note You must specify a value for this property if the Authentication Type is set to HTTPUserNamePassword. If Authentication Type is set to Auto, this property is optional.</p>
PasswordHeader	<p>Name for the password header. For this example, specify MyPassHeader. For more information about HTTP headers, see "Support for Custom HTTP and SOAP Headers" at http://go.microsoft.com/fwlink/?LinkId=106692.</p> <p> Note You must specify a value for this property if the Authentication Type is set to HTTPUserNamePassword. If Authentication Type is set to Auto, this property is optional.</p>

The following figure shows the Configure Service and Endpoint Behaviors page with the specified values.



10. On the Configure Service and Endpoint Behaviors page, click **Next**.
11. On the Configure Service Endpoint Binding and Address page, the **Select a contract to configure** box lists the contracts for the Siebel business components for which you selected the operations on the Choose Operations page. The **Operations under the selected contract** box displays the operations you selected for each artifact on the Choose Operations page.
12. In the **Configure the address and binding for the contract** box, specify values for the following:

For the property	Specify the value
Binding Configuration	<p>The wizard only supports basic HTTP binding. So, the binding configuration field is automatically populated to <code>System.ServiceModel.Configuration.BasicHttpBindingElement</code>.</p> <p>Click the ellipsis button (...) to change the properties for HTTP binding. To use a secure communication channel, you must always set the Mode property to Transport. The wizard sets the default value for the Mode property as Transport.</p> <p>For more information about the other bindings exposed, see "BasicHttpBindingElement Members" at http://go.microsoft.com/fwlink/?LinkId=103773.</p>
Endpoint Name	Specify an endpoint name for the contract.

The other fields on this page are automatically populated based on the values you specified in the earlier pages.

Click **Apply**. Perform this step for all the contracts displayed under the **Select a contract to configure** box.

Note

If you do not specify any values on this page, the default values are accepted for all the contracts.

The following figure shows the Configure Service Endpoint Binding and Address page with the specified values.

The screenshot shows the 'WCF Adapter Service Development Wizard' window. The left sidebar contains a navigation pane with the following steps: Introduction, Choose Operations, Configure Service and Endpoint Behaviors, **Configure Service Endpoint Binding and Address** (highlighted), and Summary. The main area is titled 'Configure the service endpoint binding and address.' It features two sections: 'Select a contract to configure:' with a list box containing 'BusinessObjects_Account_Account_Operation' (checked), and 'Operations under the selected contract BusinessObjects_...' with a list box containing 'Query'. Below these is a section titled 'Configure the address and binding for the contract BusinessObjects_Account_Account_Operation' containing a table with expandable sections: 'Binding Properties' (BindingConfiguration: System.ServiceModel.Configuration.BasicHttpBinding), 'Endpoint Details' (EndpointBehaviorConfiguration: EndpointBehavior1, EndpointName: AccountEndPoint), and 'Service Details' (ServiceBehaviorConfiguration: ServiceBehavior1, ServiceName: BusinessObjects_Account_Account_OperationClient). At the bottom of this section is an 'EndpointName' field with the label 'Endpoint name'. An 'Apply' button is located below the table. At the bottom of the wizard are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

Binding Properties	
BindingConfiguration	System.ServiceModel.Configuration.BasicHttpBinding

Endpoint Details	
EndpointBehaviorConfiguration	EndpointBehavior1
EndpointName	AccountEndPoint

Service Details	
ServiceBehaviorConfiguration	ServiceBehavior1
ServiceName	BusinessObjects_Account_Account_OperationClient

EndpointName
Endpoint name

13. On the Configure Service Endpoint Binding and Address page, click **Next**. The Summary page lists a tree structure of the contracts for the selected Siebel business components and, under that, the operations selected for each business component.
14. Review the summary, and then click **Finish**.
15. The wizard creates a WCF service and adds the following files to the Visual Studio project:
 - a. .svc file. This is the WCF service file. The wizard generates one file for each contract.
 - b. Web.config file.
 - c. Service code (.cs file).
16. Publish the WCF service.
 - a. Make sure SSL is enabled for Internet Information Services (IIS). For instructions on how to enable SSL for IIS, see <http://go.microsoft.com/fwlink/?LinkId=197170>.
 - b. Right-click the project in Solution Explorer, and then click **Publish**.

- c. In the **Publish Web** dialog box, specify a URL for the WCF service. For example:

```
https://<computer_name>/Siebel_Account/
```

- d. From the **Copy** box, click **All project files**.

- e. Click **Publish**.

17. Verify that the WCF service is published successfully.

- Start the IIS Microsoft Management Console. Click **Start**, point to **Administrative Tools**, and then click **Internet Information Services**.
- Navigate to the node where you published the service. For the **Siebel_Account** service, navigate to **Internet Information Services** > <Computer Name> > **Web Sites** > **Default Web Site** > **Siebel_Account**.
- On the right pane, right-click the BusinessObjects_Account_Account_Operation.svc file, and then click **Browse**.
- The Web page shows up with the URL for retrieving the WSDL. You may want to test metadata retrieval using the svcutil command. For example, the command to retrieve metadata for the Siebel_Account service is:

```
svcutil.exe https://localhost/Siebel_Account/BusinessObjects_Account_Account_Operation.svc?wsdl
```

Next Steps

You now have a WCF service for the Siebel business component. Use the Business Data Catalog Definition Editor to create an application definition file for the Siebel business component operations. See [Step 2: Create an Application Definition File for Siebel Business Component Operations](#) for instructions. The application definition file identifies where the LOB data is stored and the format in which it is stored.

See Also

Other Resources

[Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#)

Step 2: Create an Application Definition File for Siebel Business Component Operations



Time to complete: 15 minutes

Objective: The Business Data Catalog exposes and incorporates data from line-of-business (LOB) applications into portals. To incorporate this data into your portal site, you must build an application definition file that Microsoft Office SharePoint Server can consume.

The Business Data Catalog Definition Editor tool enables you to create an application definition file for the Business Data Catalog. This tool automatically generates the XML for the definition file. Therefore, you do not have to manually create the file in an XML editor.

The purpose of the Microsoft Office SharePoint Server application that you are creating is to perform a Query operation on the Account business component to retrieve a list of records. To achieve this, you must complete a set of tasks in the Business Data Catalog Definition Editor. This topic provides instructions on how to perform these tasks.

Prerequisites

- You must have the Business Data Catalog Definition Editor installed as part of the Microsoft Office SharePoint Server 2007 SDK. You can download the SDK from <http://go.microsoft.com/fwlink/?LinkId=104130>.
- You should have published the WCF service, as described in [Step 1: Publish the Siebel Business Component Operations as a WCF Service](#).

Creating an Application Definition File

This section provides step-by-step instructions to create an application definition file for the WCF service.

Connect to the WCF Service, and Create Entities

You must connect to the WCF service to extract the Web Services Description Language (WSDL) for the service. From the WSDL, the Business Data Catalog Definition Editor extracts the methods. These methods can be used to create entities. For this example, you must create one entity for the Query operation on the Account business component.

To connect to the WCF service, and create entities

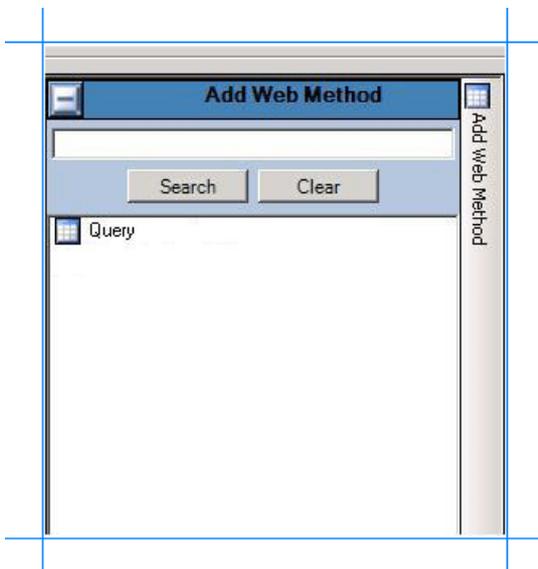
1. Start the Business Data Catalog Definition Editor. On the **Start** menu, click **Microsoft Business Data Catalog Definition Editor**.
2. In the tool, click **Add LOB System**.
3. In the Add LOB System window, click **Connect to Webservice**.
4. In the URL box, type the URL for the WCF service. The URL must be in the following format:

```
https://<computer_name>/Siebel_Account/BusinessObjects_Account_Account_Operation.svc?wsdl
```

where, BusinessObjects_Account_Account_Operation.svc is the service file created for the Siebel contract.

The URL that you must type is available when you test whether the WCF service is published successfully, as described in [Step 1: Publish the Siebel Business Component Operations as a WCF Service](#).

5. Click **Connect**.
6. Click the **Add Web Method** tab to see the operations you selected in the WCF Adapter Service Development Wizard. You will see the **Query** method.

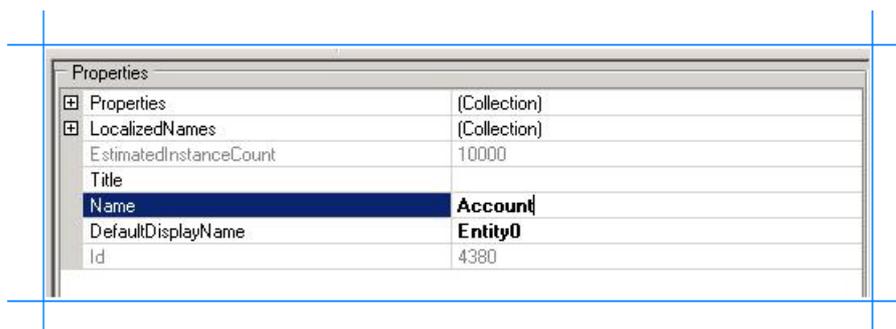


7. Drag the **Query** method to the design surface and then click **OK**.
8. In the **Enter the name for the LOB System** dialog box, type a name in the **LOB System Name** box. For this example, type **Siebel_Account**, and then click **OK**. An entity, **Entity0**, is created in the Business Data Catalog Definition Editor.

◆Important

The Business Data Catalog Definition Editor tool does not handle enumerated data types. So, the Business Data Catalog Definition Editor tool imports the fields till it encounters an enumerated data type and ignores the remaining fields. The Business Data Catalog Definition Editor tool also gives an error. You can ignore this error and proceed by clicking OK. You can manually add the required fields in the application definition file at a later stage.

9. Change the entity names to use more friendly names. For this example, change **Entity0** to **Account**.
 - a. Expand the **Siebel_Account** node, and then expand the **Entities** node.
 - b. Select the **Entity0** node.
 - c. In the Properties pane, type **Account** in the **Name** field.

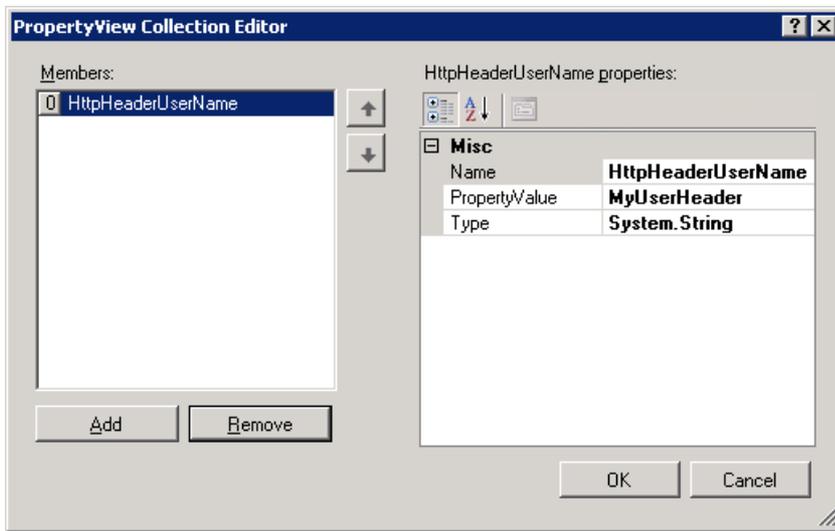


Specify User Name and Password Headers for Methods

When creating a WCF service for the selected business component operations in the Siebel system, you specified user name and password headers as part of the endpoint behavior configuration ([Step 1: Publish the Siebel Business Component Operations as a WCF Service](#)). You must specify the same values for the method properties.

To specify user name and password headers for the Query method

1. In the Metadata Objects pane, expand the **Account** node, and then expand the **Methods** node.
2. Click the **Query** node, and in the Properties pane click the ellipsis (...) button against the **Properties** field.
3. In the PropertyView Collection Editor dialog box, click **Add**, and in the Properties pane, type **HTTPHeaderUserName** for the **Name** field. Similarly, type **MyUserHeader** for the **PropertyValue** field. Select **System.String** for the **Type** field.



4. In the PropertyView Collection Editor window, click **Add**, and in the Properties pane, type **HttpHeaderPassword** for the **Name** field. Similarly, type **MyPassHeader** for the **PropertyValue** field. Select **System.String** for the **Type** field.
5. Click **OK**.

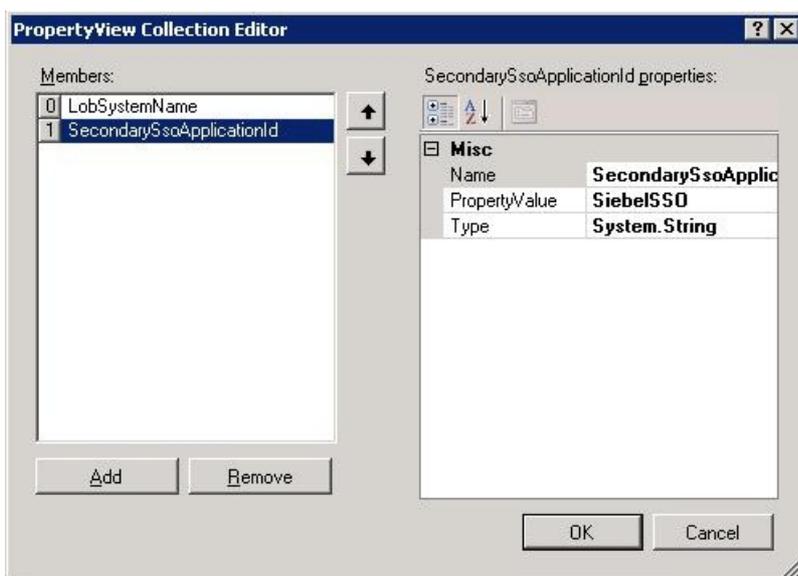
Set up Single Sign-On for Connecting to a Siebel System

After you have finished performing all the procedures in this topic, you will have created an application definition XML that can be imported into a SharePoint application. From the application, you will invoke the Siebel business component operations (exposed as Web methods) to retrieve relevant data from the Siebel system. To enable this, you must create a mapping between a user in the Siebel system and the user in the SharePoint application. You create this mapping in SharePoint Central Administration Web site after you have imported the application definition XML.

However, to create the mapping you must set a property **SecondarySsoApplicationId** in the Business Data Catalog Definition Editor.

To set the SecondarySsoApplicationId property

1. In the Metadata Objects pane, expand the **Siebel_Account** node, and then expand the **Instances** node.
2. Click **Siebel_Account_Instance** and in the Properties pane click the ellipsis (...) button against the **Properties** field.
3. In the PropertyView Collection Editor window, click **Add**, and in the Properties pane, type **SecondarySsoApplicationId** for the **Name** field. Similarly, type **SiebelSSO** for the **PropertyValue** field. Select **System.String** for the **Type** field.



4. Click **OK**.

Requirement: Perform a Query Operation on the Account Business Component

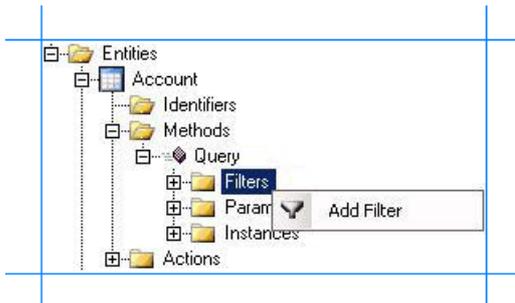
The first requirement of this example is to create an application definition that can be used to perform a Query operation on the Account business component. To achieve this requirement, you must perform the following set of tasks:

- In the Query method, create a filter, and map it to the parameter on which the Query operation is performed. For the Account business component, you will perform a query using the **SearchExpr** parameter. So, you will map the filter to the **SearchExpr** parameter.
- Create a Finder method instance for the Query method. A Finder method retrieves a list of records based on a filter.

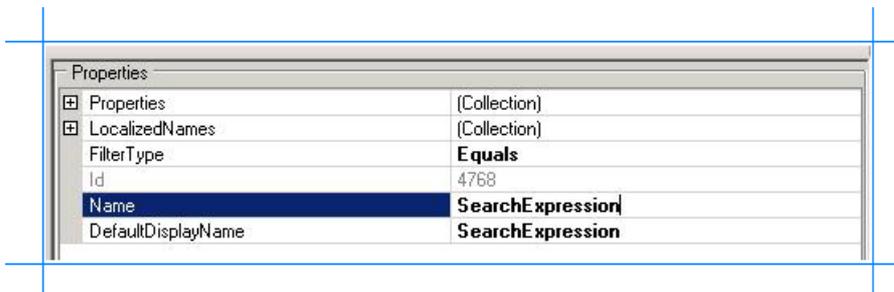
To create a filter, and map it to the SearchExpr parameter

1. Create a filter for the Query method.

- In the Metadata Objects pane, expand the **Account** node, and then expand the **Methods** node.
- Expand the Query method, right-click **Filters**, and then click **Add Filter**.

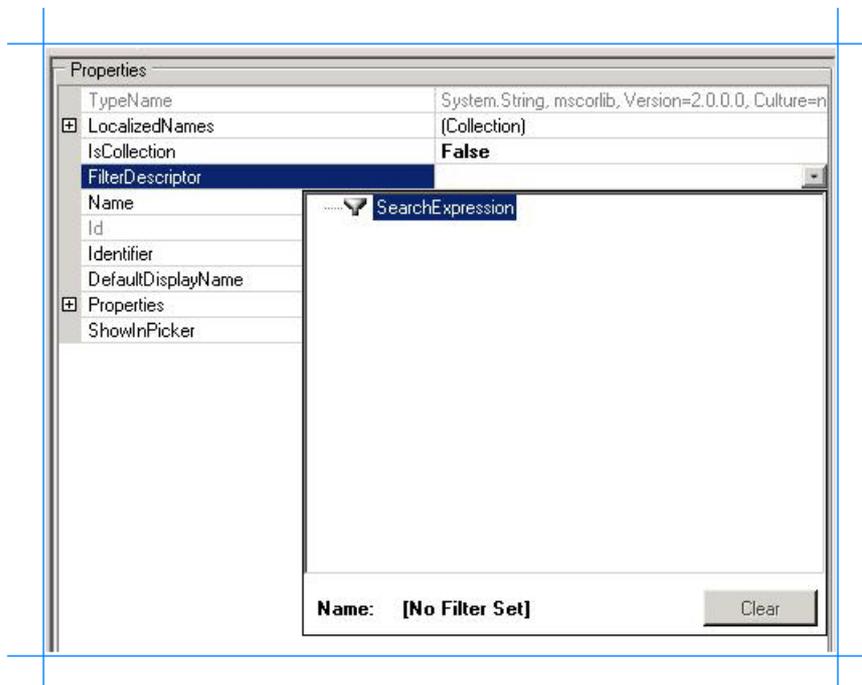


- In the Properties pane, type **SearchExpression** for the **Name** field.
- For the **FilterType** property, select **Equals**.



2. Map the filter to the **SearchExpr** parameter in the Query method.

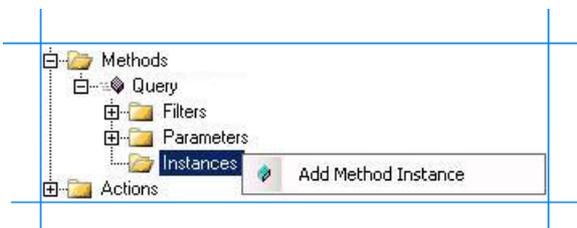
- In the Metadata Objects pane, expand the **Account** node, and then expand the **Methods** node.
- Expand the Query method, and then expand the **Parameters** node.
- Expand the **AccountQueryInputRecord** node, and then expand the second **AccountQueryInputRecord** node.
- Click the **SearchExpr** node and in the Properties pane, select **SearchExpression** from the **FilterDescriptor** list.



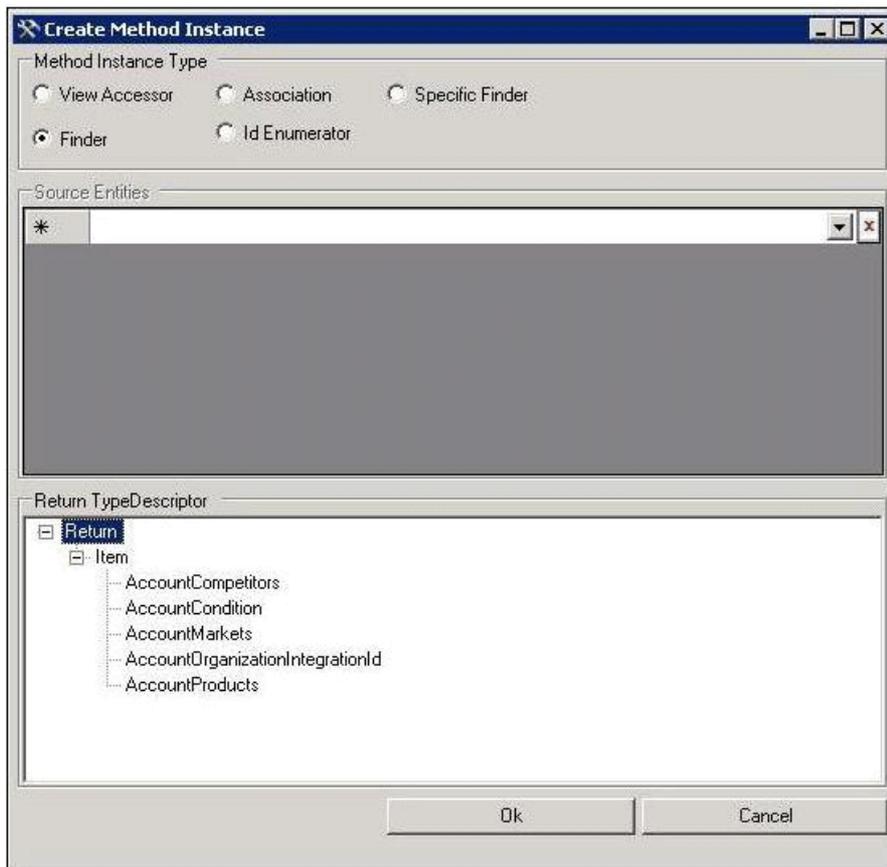
Important
 The **AccountQueryInputRecord** also contains a **QueryFields** node, which in turn contains an **Item** node. You must delete the **Item** node, otherwise the Query operation on the Account business component might not give the desired results. To delete the **Item** node, right-click the node, and then select **Delete**.

To create a Finder method instance for Query method

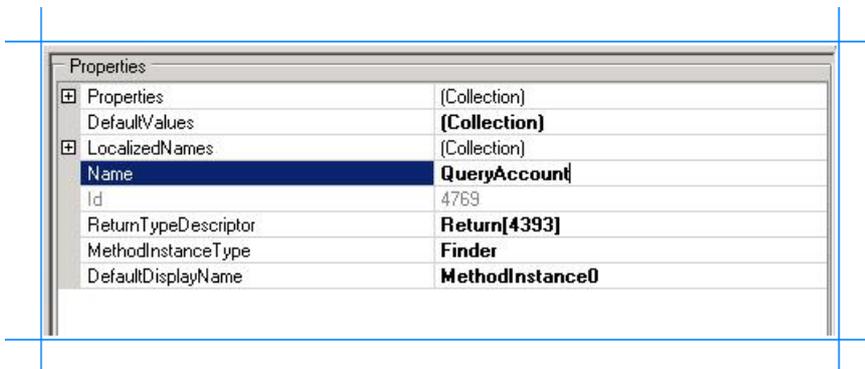
1. In the Metadata Objects pane, expand the **Account** node, and then expand the **Methods** node.
2. Expand the **Query** node, right-click **Instances**, and then click **Add Method Instance** to open the Create Method Instance window.



3. In the Create Method Instance window, click **Finder** for the **Method Instance Type**.
4. Click **Return** from **Return TypeDescriptor** section.



5. Click **OK**.
6. In the Properties pane, type **QueryAccount** for the **Name** field.



Remove the Parameters of System.Nullable Type

The return parameters for Query function may contain parameters that are of System.Nullable type. Due to the presence of these parameters in the application definition, you might get an error while presenting Siebel data on a SharePoint portal. So, you must remove the parameters of System.Nullable type from the application definition.

Also, for each parameter of System.Nullable type, the Business Data Catalog Definition Editor creates another parameter of System.Boolean type, and appends "Specified" to the parameter name. For example, the parameter AccountRole is of System.Nullable type. So, the Business Data Catalog Definition Editor adds an AccountRoleSpecified parameter to the list of parameters. You must remove such parameters as well.

Note

You can see the parameter type by selecting the parameter in the Business Data Catalog Definition Editor, and looking at the value for the **TypeName** property in the Properties pane.

Note

You can skip this step if the application does not contain any parameters of System.Nullable type.

To remove the parameters of System.Nullable type for the Query method

1. In the Metadata Objects pane, expand the **Account** node, and then expand the **Methods** node.
2. Expand the **Query** node, and then expand the **Parameters** node.
3. Expand the **Return** node, and then expand the second **Return** node.
4. Right-click the parameter you want to delete, and then select **Delete**.
5. In the dialog box, click **OK**.

Export the Application Definition to a File

You have now created an application definition that contains the Siebel system instance metadata. You must export this definition to an XML file, which can be imported into Microsoft Office SharePoint Server.

To export the application definition to a file

1. Right-click the **Siebel_Account** node in the Metadata Objects pane, and then click **Export**.
2. Save the file as Siebel_Account.xml.

Modify the Application Definition File to Include Specific Parameters

As mentioned earlier in this topic, the Business Data Catalog Definition Editor tool does not handle enumerated data types. The Business Data Catalog Definition Editor tool imports the fields till it encounters an enumerated data type and ignores the remaining fields. So, certain fields that you want in your application might be omitted. You can manually edit the application definition file to include those fields.

Note

You must make sure the parameters you are adding are present in the .cs file generated by the WCF Adapter Service Development Wizard in [Step 1: Publish the Siebel Business Component Operations as a WCF Service](#).

In this application definition file, you will add or remove return parameters for the **QueryAccount** method.

To modify the application definition file

1. Open the application definition file, Siebel_Account.xml, by using Visual Studio or any other editor.
2. Modify the application definition file to replace the parameters for the **QueryAccount** method.
 - a. Within the application definition file, search for the following:

```
<TypeDescriptor TypeName="BDC.AccountQueryRecord,Siebel_Account" Name="Item">
```

- b. Within the <TypeDescriptors> tag, replace the existing <TypeDescriptor> elements with the following:

```
<TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=<token>" Name="Id" />
<TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=<token>" Name="Country" />
<TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=<token>" Name="Name" />
<TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=<token>" Name="Location" />
```

- c. Save and close the file.

Tip

You can import the updated application definition file back in the Business Data Catalog Definition Editor tool to see the newly added fields. However, before importing you will have to remove the existing "Siebel_Account" application from the Business Data Catalog Definition Editor tool.

Next Steps

You must now create a SharePoint application to retrieve data from a Siebel system. See [Step 3: Create a SharePoint Application to Retrieve Data from Siebel](#) for instructions.

See Also

Other Resources

[Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#)

Step 3: Create a SharePoint Application to Retrieve Data from Siebel



Time to complete: 15 minutes.

Objective: You must now take the application definition file you created by using the Business Data Catalog Definition Editor, and import it into Office SharePoint Server.

Prerequisites

- You should have created an application definition file, as described in [Step 2: Create an Application Definition File for Siebel Business Component Operations](#).
- The Microsoft Single Sign-on service must be running.

How to Create a SharePoint Application

Creating a SharePoint application involves the following steps:

- Create a single sign-on (SSO) application in SharePoint
- Create a Shared Services Provider (SSP)
- Import the application definition file
- Create a Web Part page, and add Web Parts

Creating an SSO Application in SharePoint

To access the data in a Siebel system from a SharePoint application, you must set up an SSO application that maps a SharePoint user to a Siebel user. Creating an SSO application in SharePoint involves the following steps:

1. **Manage server settings for single sign-on.** In this step, you specify a user account that can manage and set up the single sign-on service. You can do so from the Manage Server Settings page. This option is available from the SharePoint Central Administration console. For more information about this step, refer to the "Configure Single Sign-On for Office SharePoint Server 2007" section at <http://go.microsoft.com/fwlink/?LinkId=105291>.
2. **Manage settings for enterprise application definitions.** In this step, you configure the settings for the enterprise application definition. You can do so from the Manage Settings for Enterprise Application Definitions page. This option is available from the SharePoint Central Administration console.
 - a. On Central Administration, on the top navigation bar, click **Operations**.
 - b. On the Operations page, in the **Security Configuration** section, click **Manage settings for single sign-on**.
 - c. On the Manage Settings for Single Sign-On page, in the **Enterprise Application Definition Settings** section, click **Manage settings for enterprise application definitions**.
 - d. On the Manage Enterprise Application Definitions page, provide values for the **Display name**, **Application name**, and the **Contact e-mail address** fields.

◆ Important

For the **Application name** field, make sure you specify the same SSO application name that you specified for the **SecondarySsoApplicationId** variable while creating the application definition file, as described in [Step 2: Create an Application Definition File for Siebel Business Component Operations](#).

e. Leave the other fields as default, and click **OK**.

3. **Manage account information for enterprise application definitions.** In this step, you enable individual users or groups to connect to an enterprise application from SharePoint. Essentially, in this step you map an individual user or group to a user in the LOB system. You also specify the credentials to connect to the LOB system. You can do so from Manage Account Information for Enterprise Application Definitions page. This option is available from the SharePoint Central Administration console. For more information about this step, refer to the "Manage account information for an enterprise application definition" section at <http://go.microsoft.com/fwlink/?LinkId=105291>.

Creating a Shared Services Provider

An SSP is a logical grouping of shared services and their supporting resources. You can create an SSP using the SharePoint Central Administration console.

You must define a Web site while creating an SSP. Remember the port number and the site address you create. You will import the Business Data Catalog application definition to this site.

For more information about creating an SSP, see "Chapter overview: Create and configure Shared Services Providers" at <http://go.microsoft.com/fwlink/?LinkId=105119>.

Importing the Application Definition File

You must now import the application definition file into the SSP.

To import the application definition file

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP to which you want to import the application definition.
3. In the **Business Data Catalog** section, click **Import application definition**.
4. On the Import Application Definition page that opens, browse to Siebel_Account.xml, select the file, and then click **Open**.
5. Click **Import**.
6. Click **OK**.

After importing the application, you can see your application by going to the **View Applications** link. Click the application name to see the entities in the application.

Creating Web Parts

You must now create Web Parts in your SharePoint site to view and manage the business data that will be extracted from the Siebel system. Web Parts are reusable components that can contain any kind of Web-based information, including analytical, collaborative, and database information.

In this tutorial, Web Parts are created for the method instances that were created in Business Data Catalog Definition Editor. Office SharePoint Server provides different kinds of Web Parts for specific use. For the Finder method instance, we will use the **Business Data List** Web Part. This Web Part enables you to specify a search expression to perform a query on the Account business component. For this tutorial, we call this the **Query Accounts** Web Part.

This section provides instructions to create these Web Parts. For more information about creating Web Parts, see the Microsoft Office SharePoint Server 2007 document ("Customize business data lists, Web Parts, and sites") at <http://go.microsoft.com/fwlink/?LinkId=104131>.

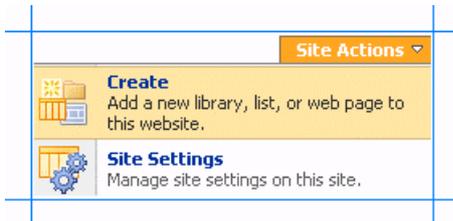
The Web Parts will be added to a single Web Part page. You must create a Web Part page before adding the Web Parts. For this tutorial, the Web Part page is called **Siebel Account**.

Creating a Web Part Page

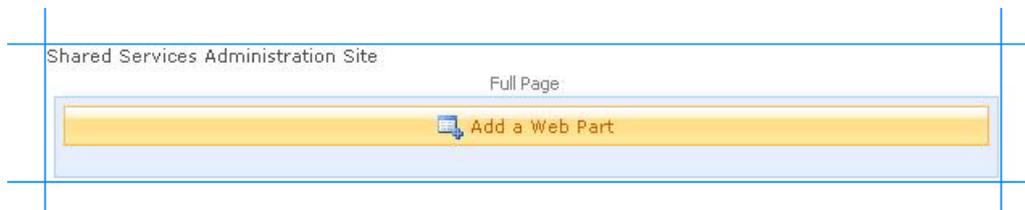
This section provides instructions to create a Web Part page.

To create a web part page

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP to which you want to import the application definition.
3. On the Shared Services Administration page, from the top right-hand corner, click **Site Actions**, and then click **Create**.



4. On the Create page, under the **Web Pages** section, click **Web Part Page**.
5. In the New Web Part page, do the following:
 - a. In the **Name** field, specify a name for the page. For this tutorial, specify the name as **Siebel Account**.
 - b. Select the **Overwrite if file already exists** check box, if you want to overwrite old pages with the same name as the page you create.
 - c. In the **Layout** section, from the **Choose a Layout Template** box, select a layout for the Web Part page. For this tutorial, select **Full Page, Vertical**.
 - d. In the **Save Location** section, in the **Document Library** list, select **Form Templates**.
 - e. Click **Create**. The following figure shows a Web Part page after it is just created.



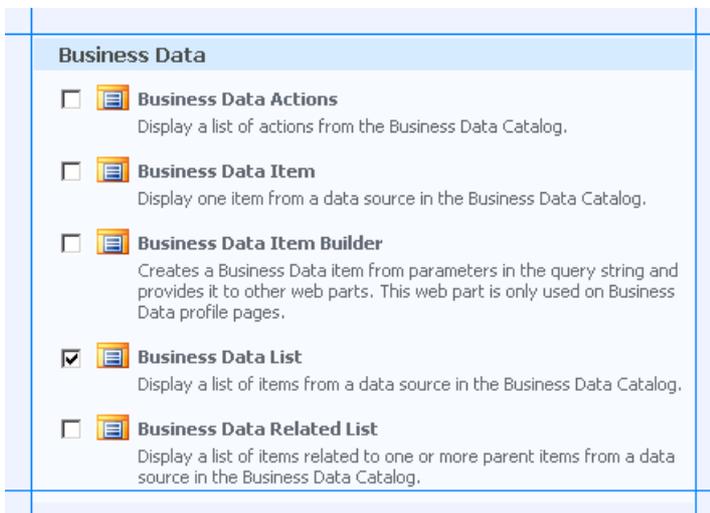
You must now add the different Web Parts to this page.

Adding a Business Data List Web Part

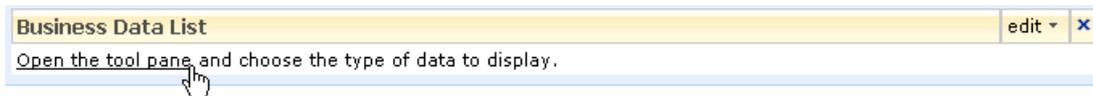
You must now add a Business Data List Web Part to the Web Part page. Using this Web Part, you will query the Account business component using a search expression. This Web Part corresponds to the Finder method instance (QueryAccount) you created in the Business Data Catalog Definition Editor.

To add a Business Data List Web Part

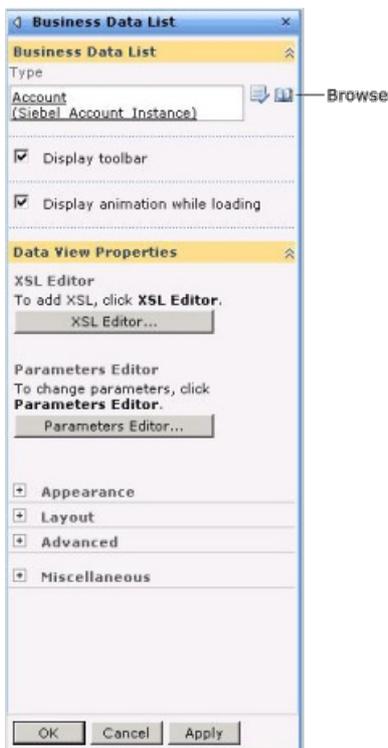
1. In the **Siebel Account** page, in the **Header** section, click **Add a Web Part**.
2. In the **Add Web Parts** dialog box, in the **Business Data** section, select the **Business Data List** check box, and then click **Add**.



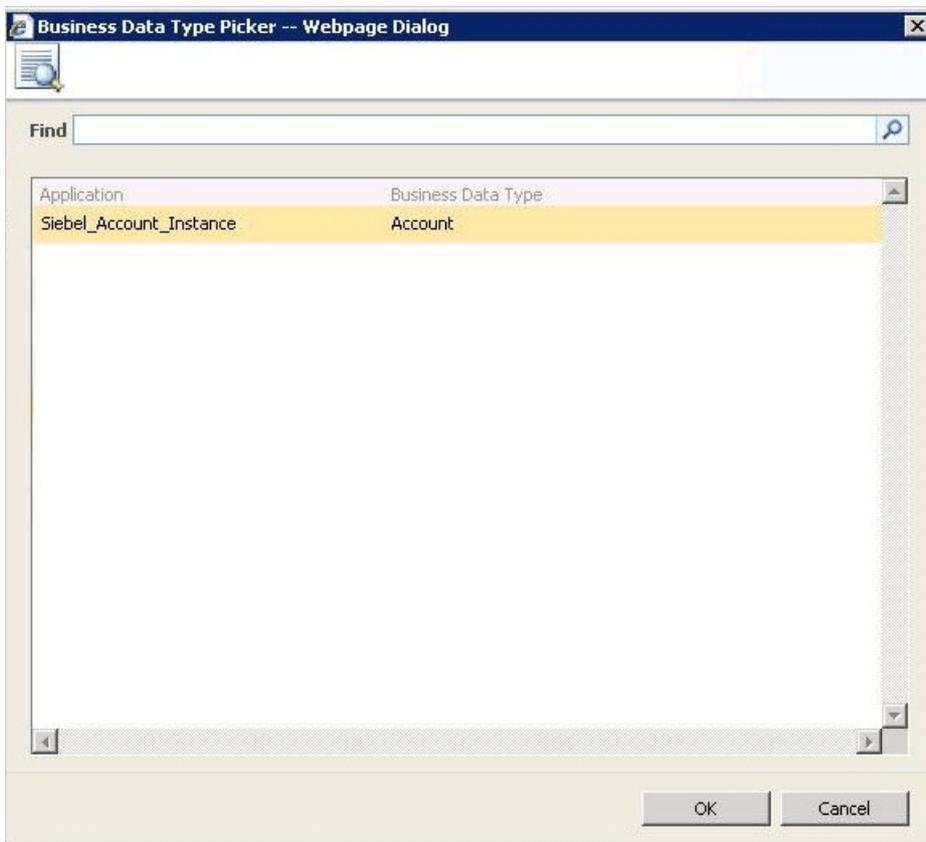
- In the newly added Business Data List Web Part, click the **Open the tool pane** link.



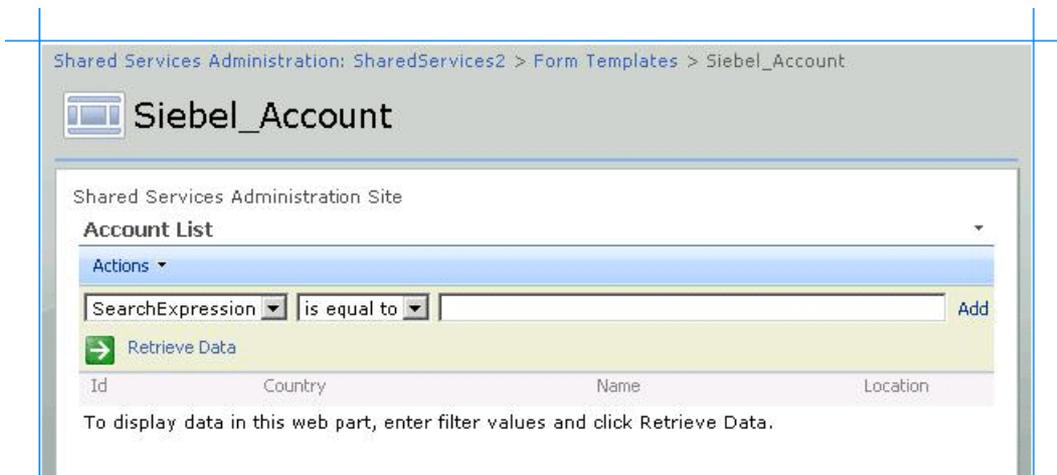
- The Business Data List tool pane opens in the right pane. In the **Business Data List** section, for the **Type** field, click the **Browse** button.



- In the **Business Data Type Picker** dialog box, select the **Siebel_Account_Instance** application, and then click **OK**.



6. Expand the **Appearance** node, and in the **Title** field, specify a title for the Web Part. For this Web Part, specify **Account List**.
7. In the Business Data List tool pane, click **Apply**, and then click **OK**. The Business Data List Web Part now looks like the following:



Note
 You can also change the order in which the parameter columns appear. You can do so by clicking the **Edit View** link towards the right corner of the Web Part.

8. Click **Exit Edit Mode** from the top right corner of the page.

Next Steps

Test the SharePoint application by retrieving data from a Siebel system. See [Step 4: Test Your SharePoint Application](#).

See Also

Other Resources

[Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#)

Step 4: Test Your SharePoint Application



Time to complete: 5 minutes.

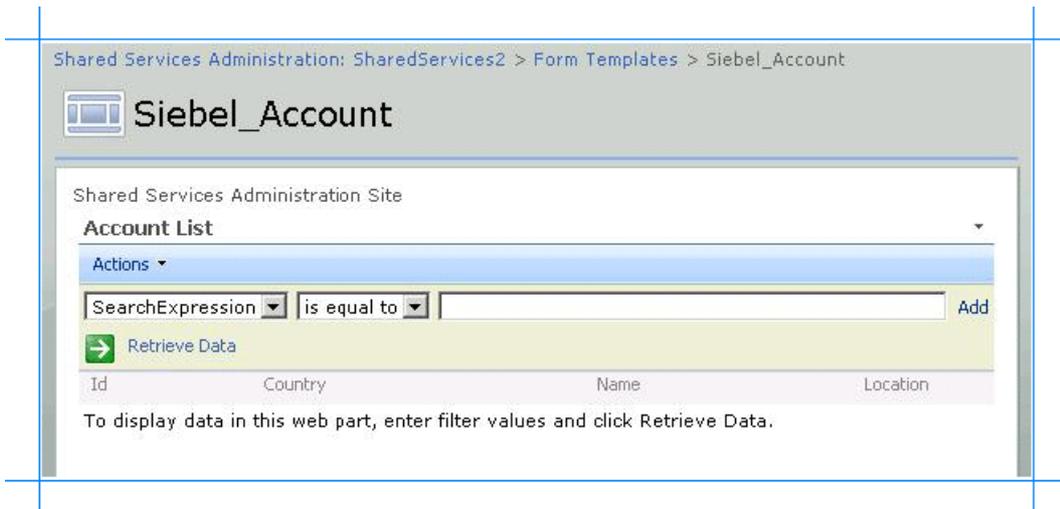
Objective: After you have added Web Parts in the SharePoint site and created an application, you must test the application by retrieving some data from the Siebel system. This section provides instructions on how to use the application to retrieve the data from the Siebel system.

Prerequisites

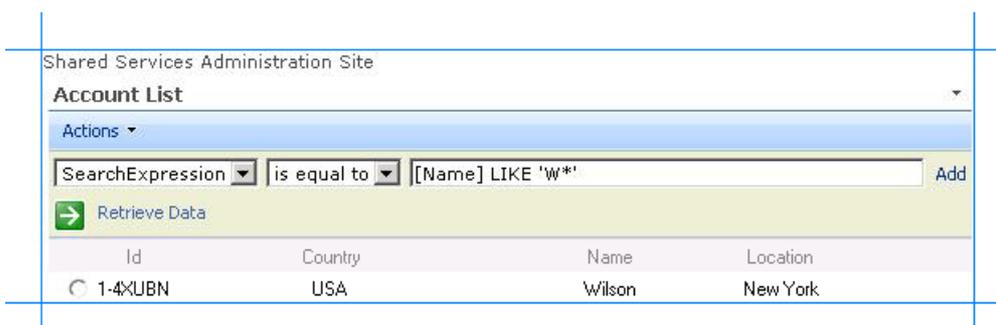
You should have created the Web Part page containing the appropriate Web Parts to retrieve business data. See [Step 3: Create a SharePoint Application to Retrieve Data from Siebel](#).

To test the SharePoint application

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the SSP under which you created the application.
3. In the left pane, click **View All Site Content**. From the right pane, click **Form Templates**.
4. In the **Form Category** list, click **Siebel Account**. You specified this name while creating the Web Part page. The following figure shows the Web Part page that you created.



5. Query the Account business component based on a search string. For example, specify the search expression `[Name] LIKE 'W*'`. To do so:
 - a. In the **Account List** section, from the first list, select **SearchExpression**, and then select **is equal to**.
 - b. In the text field, type **[Name] LIKE 'W*'**.
 - c. Click **Retrieve Data** link, or press ENTER. The following figure shows the records retrieved from the Siebel system that satisfy the search criteria.



See Also

Other Resources

[Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#)

Tutorial 2: Migrating BizTalk Projects

The previous version of the Siebel adapter that shipped with Microsoft BizTalk Server differs from the WCF-based Siebel adapter in many aspects, including:

- The design-time experience of creating a BizTalk project.
- The metadata retrieval experience.
- Schema file name and namespace.
- Data type mappings.
- The operations that can be performed using the adapter.
- Physical port configuration in the BizTalk Server Administration console

These differences are explained in the topics within [Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#).

However, you can make changes to the BizTalk project created using the previous version of the adapter and make it work with the WCF-based Siebel adapter.

This tutorial provides instructions on the changes you should make to the existing BizTalk project created using the previous version of the adapter.

Note

In this tutorial, for the sake of brevity, the previous version of the Siebel adapter will be referred to as vPrev Siebel adapter. Similarly, a BizTalk project that uses the vPrev Siebel adapter will be referred to as vPrev BizTalk project.

Sample Used for the Tutorial

This tutorial is based upon a sample (Siebel_BussComp_Migration) that demonstrates how to migrate a vPrev BizTalk project that performs an Insert operation on the Account Siebel business component. The sample is provided with Microsoft BizTalk Adapter Pack. For more information, see [Samples](#).

Prerequisites

- You must have a vPrev BizTalk project. This tutorial involves a BizTalk project that performs an Insert operation on the Account business component.
- You must have a request message to perform an Insert operation on the Account business component using the vPrev Siebel adapter. The request message must conform to the schema of the Insert operation generated using the vPrev Siebel adapter.
- You must have completed the steps in [Before You Develop BizTalk Applications](#).

Understanding a BizTalk Project Created Using the Previous Version of the Adapter

The key constituents of a vPrev BizTalk project created are:

- **BizTalk orchestration.** This is a simple orchestration that picks request messages from a file location, sends the request message to the Siebel system using a Siebel send-receive port, receives the response, and saves it to another file location.
- **Schema for the operation you wish to perform on the Siebel business component.** This tutorial involves a BizTalk project that performs an Insert operation on the Account business component. The schema generated for the Account business component is AccountService_Account_x5d.xsd. This schema is generated using the vPrev Siebel adapter.

Note

Unlike the WCF-based Siebel adapter, the vPrev Siebel adapter does not support generating metadata for specific operations on a business component. By default, the adapter generates schema for all the operations supported on the business component. For more such differences between the vPrev Siebel adapter and the WCF-based Siebel adapter, see [Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#).

- **Request message.** The request message to perform an Insert operation on the Account business component. The schema of the request message conforms to the schema of the Insert operation as surfaced by the vPrev Siebel adapter.

How to Migrate a BizTalk Project Created Using the Previous Version of the Adapter

The goal of this migration tutorial is to enable you to send a request message, which conforms to schema generated by the vPrev Siebel adapter, using a WCF-Custom port that can only process messages conforming to the WCF-based Siebel adapter. So, in short, the migration exercise involves configuring the WCF-Custom port to process messages that do not conform to the WCF-based Siebel adapter's schema.

However, to be able to configure the WCF-Custom port appropriately, you must perform the following tasks:

- Generate metadata for the Insert operation on the Account business component using the WCF-based Siebel adapter.
- Map the request message for performing an Insert operation using the vPrev Siebel adapter to a request message for performing an Insert operation using the WCF-based Siebel adapter.
- Map the response message received using the WCF-based Siebel adapter to the response message for the vPrev Siebel adapter.
- Create a WCF-Custom Siebel send-receive port in the BizTalk Server Administration console.
- Configure the WCF-Custom port to use the request and response mappings.

In This Section

- [Step 1: Modify the vPrev BizTalk Project](#)
- [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#)
- [Step 3: Test the Migrated Application](#)

See Also

Other Resources

[Siebel Adapter Tutorials](#)

Step 1: Modify the vPrev BizTalk Project



Step 1

Time to complete: 10 minutes

Objective: In this step, you make the following changes to the existing vPrev BizTalk project:

- Generate metadata for the Insert operation on the Account business component using the WCF-based Siebel adapter.
- Map the request message for performing an Insert operation using the vPrev Siebel adapter to a request message for performing an Insert operation using the WCF-based Siebel adapter.
- Map the response message received using the WCF-based Siebel adapter to the response message for the vPrev Siebel adapter.

Prerequisite

- You must have a vPrev BizTalk project to perform an Insert operation on the Account business component in the Siebel system.

To modify the vPrev BizTalk project

1. Generate metadata for the Insert operation on the Account business component using the WCF-based Siebel adapter. You can use the Consume Adapter Service BizTalk Project Add-in to generate metadata.

For instructions on how to generate metadata, see [Retrieving Metadata for Siebel Operations in Visual Studio](#). After the schema is generated, a file with the name similar to *SiebelBindingSchema.xsd* is added to the BizTalk project. This file contains the schema for sending a message to perform the Insert operation on the Account business component using the WCF-based Siebel adapter.

2. Generating the metadata for the Insert operation also creates a port binding file. In the next step, this binding file will be used to create a WCF-Custom send port to send messages to the Siebel system. The SOAP action for the operation is also set to the operation for which you generated metadata. For example, if you generate metadata for the Insert operation, the operation name in the SOAP action on the send port will be "Insert". However, the operation name on the logical send port that you create as part of the orchestration could be different, for example, "Operation_1". As a result, when you send messages to the Siebel system using the send port, you get an error. To prevent this, make sure the operation name on the logical send port in your orchestration is the same as the operation name for which you generated metadata.

So, in case of this tutorial, because you generate metadata for the Insert operation, change the name of the logical send port operation to "Insert".

3. For the request message, map the schema generated using vPrev Siebel adapter to the schema generated using the WCF-based Siebel adapter.

- a. Add a BizTalk mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.

In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **RequestMap.btm**. Click **Add**.

- b. From the Source Schema pane, click **Open Source Schema**.

- c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the request message for the vPrev Siebel adapter. For this tutorial, select *Siebel_BussComp_Migration.AccountService_Account_x5d*. Click **OK**.

- d. In the **Root Node for Source Schema** dialog box, select *Insert*, and then click **OK**.

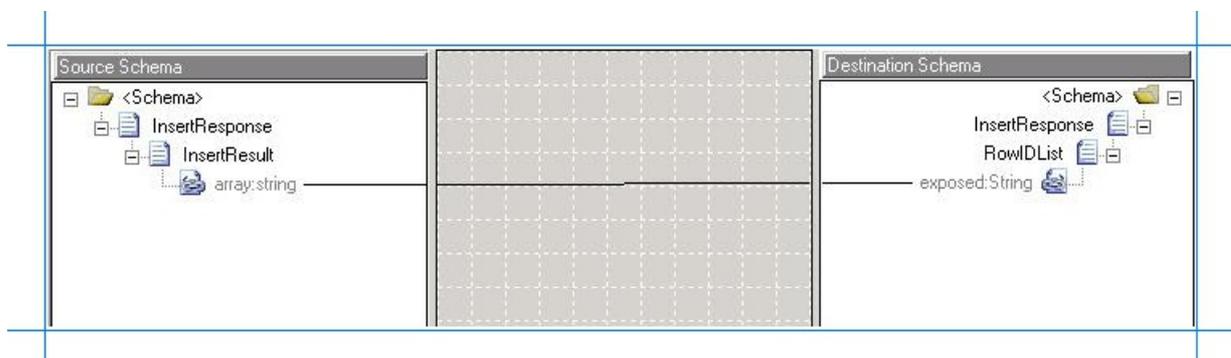
- e. From the Destination Schema pane, click **Open Destination Schema**.
- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the request message for the WCF-based Siebel adapter. For this tutorial, select *Siebel_BussComp_Migration.SiebelDBBindingSchema*, and then click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *Insert*, and then click **OK**.
- h. Map the following elements in both the schemas: **Currency_Code**, **Current_Volume**, **Customer_Account_Group**, **Location**, **Main_Phone_Number**, **Name**, **Party_Name**, **Primary_Address_Id**,
- i. Save the map.

4. For the response message, map the schema generated using the vPrev Siebel adapter to the schema generated using the WCF-based Siebel adapter.

- a. Add a BizTalk mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.

In the Add New Item dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **ResponseMap.btm**. Click **Add**.

- b. From the Source Schema pane, click **Open Source Schema**.
- c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the response message for the WCF-based Siebel adapter. For this tutorial, select *Siebel_BussComp_Migration.SiebelDBBindingSchema*. Click **OK**.
- d. In the **Root Node for Source Schema** dialog box, select *InsertResponse* and click **OK**.
- e. From the Destination Schema pane, click **Open Destination Schema**.
- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the response message for the vPrev Siebel adapter. For this tutorial, select *Siebel_BussComp_Migration.AccountService_Account_x5d*. Click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *InsertResponse* and click **OK**.
- h. Map the **array:string** element in the source schema to the **exposed:string** element in the destination schema, as illustrated in the following figure.



- i. Save the map.

- 5. Save and build the BizTalk solution. Right-click the solution, and then click **Build Solution**.
- 6. Deploy the solution. Right-click the solution, and then click **Deploy Solution**.

Next Steps

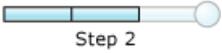
Create a WCF-custom send port and configure it to use the maps you created in this step, as described in [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#).

See Also

Concepts

[Tutorial 2: Migrating BizTalk Projects](#)

Step 2: Configure the Orchestration in BizTalk Server Administration Console



Time to complete: 10 minutes

Objective: In this step, you perform the following tasks:

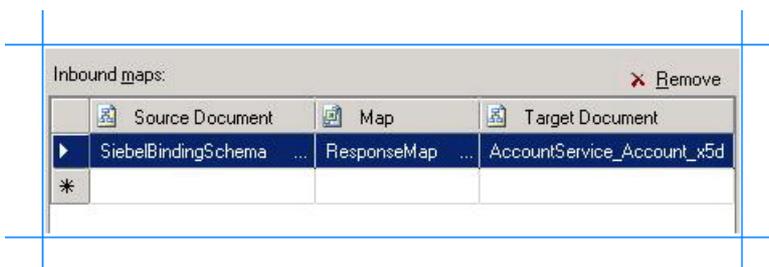
- Create a WCF-Custom send-receive port to send and receive messages from the Siebel system using the Siebel adapter. Configure this port to use the maps you created in the previous step.
- Configure the orchestration you deployed in the last step to use the WCF-Custom port.

Prerequisite

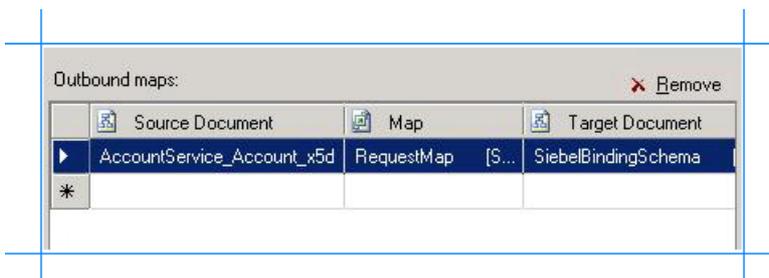
- You must have deployed the BizTalk orchestration for which you want to configure the WCF-Custom port.

To create a WCF-Custom port

1. When you generate schema for an operation on the Siebel system using Consume Adapter Service BizTalk Project Add-in, a binding file is also added to the BizTalk project. You can import this binding file into your BizTalk application to create a WCF-Custom send-receive port. For instructions on importing a binding file, see [Importing Bindings](#).
2. After you import the binding file, a send port is created under the **Send Ports** folder in the BizTalk Server Administration console.
3. Right-click the WCF-Custom port, and then click **Properties**.
4. From the left pane of the send port properties dialog box, click the **General** tab. From the right pane, click **Configure**.
5. In the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and specify the credentials to connect to a Siebel system.
6. Click **OK**.
7. From the left pane of the send port properties dialog box, click **Inbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **ResponseMap**.



8. From the left pane of the send port properties dialog box, click **Outbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **RequestMap**.



9. Click **OK**.

To configure the BizTalk application

1. In the BizTalk Server Administration console, expand **BizTalk Group**, expand **Applications**, and expand the BizTalk Application where the orchestration is deployed.
2. Right-click the BizTalk application, and then select **Configure**.
3. From the left pane, click the orchestration to configure. From the right pane, from the **Host** drop-down list, select a BizTalk host instance.
4. Under the **Bindings** box, map the logical ports of the BizTalk orchestration to the physical ports in the BizTalk Server Administration console.
 - a. Select the file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Siebel system.
 - b. Select the file port where the BizTalk orchestration will drop the response message containing the response from the Siebel system.
 - c. Select the WCF-Custom send port you created earlier in this topic.
 - d. Click **OK**.

For more information about configuring an application, see "How to Configure an Application" at <http://go.microsoft.com/fwlink/?LinkId=102360>.

Next Steps

You have now completed migration of your vPrev BizTalk project to a BizTalk project that sends messages to the Siebel system using the WCF-based Siebel adapter. You must now test the migrated BizTalk application by sending a request message to invoke the Insert operation on the Account business component, as described in [Step 3: Test the Migrated Application](#).

See Also

Concepts

[Tutorial 2: Migrating BizTalk Projects](#)

Step 3: Test the Migrated Application



Time to complete: 5 minutes

Objective: In this step, you will test the migrated application by performing an Insert operation on the Account business component. To do this, you drop a request message that conforms to the schema generated using the vPrev Siebel adapter.

Prerequisites

- Configure the BizTalk application by mapping the logical ports in the BizTalk orchestration to physical ports in the BizTalk Server Administration console.
- Configure the BizTalk application to use the WCF-Custom send port for the WCF-based Siebel adapter.

To test the migrated application

1. From the Siebel_BussComp_Migration folder, copy the AccountInsert.xml request message. This request message conforms to the schema generated by the vPrev Siebel adapter. Using the outbound map, the WCF-Custom send port converts this to conform to the schema for the WCF-based Siebel adapter and sends it to the Siebel system.

```
<Insert xmlns="http://schemas.microsoft.com/[Siebel://Business Objects/Account/Account
 ]">
  <AccountInsertRecordSet>
    <AccountInsertRecord xmlns="http://schemas.microsoft.com/Business_Objects">
      <Currency_Code>USD</Currency_Code>
      <Customer_Account_Group>Sold-To-Party</Customer_Account_Group>
      <Location>Location_1</Location>
      <Main_Phone_Number>012345678</Main_Phone_Number>
      <Name>John_Smith</Name>
      <Party_Name>Party_Name_1</Party_Name>
      <Primary_Address_Id></Primary_Address_Id>
    </AccountInsertRecord>
  </AccountInsertRecordSet>
</Insert>
```

2. Paste the request message to the folder mapped to the file receive location.
3. The orchestration consumes the request message and sends it to the Siebel system. The response from the Siebel system is received in the schema that conforms with the schema of the WCF-based Siebel adapter. Using the inbound map, the WCF-Custom send port converts this to the schema for the vPrev Siebel adapter. The response from the Siebel system is saved to the other file location defined as part of the orchestration. The response for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<ns0:InsertResponse xmlns:ns0="http://schemas.microsoft.com/[Siebel://Business Objects
 /Account/Account]" xmlns:exposed="http://schemas.microsoft.com" xmlns:Business_Objects
 ="http://schemas.microsoft.com/Business_Objects">
  <ns0:RowIDList>
    <exposed:String>1-8EWWZ</exposed:String>
  </ns0:RowIDList>
</ns0:InsertResponse>
```

See Also

Concepts

[Tutorial 2: Migrating BizTalk Projects](#)

Community Resources

There are two principal sources of information from Microsoft about the BizTalk Adapter Pack:

- The documentation and samples installed with the BizTalk Adapter Pack.
- The Microsoft BizTalk Adapter Pack forums.

BizTalk Adapter Pack Documentation

You can install the BizTalk Adapter Pack documentation when you install the BizTalk Adapter Pack. (For information about how to install the BizTalk Adapter Pack, see the BizTalk Adapter Pack installation guide. The installation guide is typically available under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.) The core documentation covers the concepts and procedures required to use the BizTalk Adapter Pack.

After you install the BizTalk Adapter Pack, you can access the documentation through the **Microsoft BizTalk Adapter Pack** program group on the Start menu.

BizTalk Adapter Pack Samples

This BizTalk Adapter Pack release provides samples demonstrating how to use the adapters with:

- BizTalk Server.
- The Windows Communication Foundation (WCF) channel model.
- The WCF service model.
- The .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel).

In addition, migration samples are provided.

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>.

BizTalk Adapter Pack Forums

You can post your queries about the BizTalk Adapter Pack on the forums.

Resource	Location
BizTalk Adapter Pack Forums	http://go.microsoft.com/fwlink/?LinkId=87695

Getting Assistance with the BizTalk Adapter Pack

Should you encounter any issues with the BizTalk Adapter Pack, you must mention the product ID (PID) for the BizTalk Adapter Pack in all communications with Microsoft Support. The PID is available in a file, pid.txt, installed by the setup wizard, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

The end-user license agreement (EULA) for the BizTalk Adapter Pack is available in a file, EULA.rtf, installed by the setup wizard, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack.

See Also

Other Resources

[Getting Started](#)

Frequently Asked Questions

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for Siebel eBusiness Applications.

What are the supported Siebel server versions?

The Siebel adapter can be used to connect to Siebel system with the following versions:

- Siebel 7.5 (7.5.3.17 + Quick Fix 7.5.3.17[16285]QF0H05)
- Siebel 7.7 (7.7.2.10)
- Siebel 7.8 (7.8.2.8)
- Siebel 8.0 (8.0.0.2)

What are the supported Siebel client versions?

The Siebel adapter supports the following Siebel client version:

- Siebel Web Client 7.5
- Siebel Web Client 7.7
- Siebel Web Client 7.8
- Siebel Web Client 8.0

What are the supported BizTalk versions for the BizTalk Adapter Pack?

The BizTalk Adapter Pack is supported with Microsoft BizTalk Server 2010.

In which user context should the setup be run?

You must always run the BizTalk Adapter Pack setup wizard under a user context with administrative privileges on the computer.

Does the Microsoft BizTalk Adapter Pack support tracing?

Microsoft BizTalk Adapter Pack enables adapter clients to activate WCF tracing and adapter specific tracing. For more information about tracing see [Diagnostic Tracing and Message Logging](#).

Where are the trace files stored?

The trace files are stored at a location specified by the adapter clients. The adapter clients can specify a location for storing the files while enabling tracing. For more information about tracing see [Diagnostic Tracing and Message Logging](#).

How do I view the traces?

You can use the Windows Communication Foundation (WCF) Service Trace Viewer tool to view the traces. For more information about the tool see <http://go.microsoft.com/fwlink/?LinkId=91243>.

Are performance counters available for the adapters?

The BizTalk Adapter Pack provides a "LOB Time (Cumulative)" performance counter to measure the time, in milliseconds, that the LOB client library takes to complete an action that the adapter initiates. For more information about performance counters see [Using Performance Counters](#).

How do I enable performance counters for adapters?

The performance counters can be enabled or disabled by setting the binding property **EnablePerformanceCounters**. Set **EnablePerformanceCounters** binding property to **True** to enable performance counters. To disable performance counters, set **EnablePerformanceCounters** to **False**. By default, **EnablePerformanceCounters** is set to **False**.

See Also

Other Resources

[Getting Started](#)

Planning and Architecture

This section describes:

- The architecture of end-to-end solutions that use the Siebel adapter to operate on a Siebel system.
- The internal architecture of the Siebel adapter.

An understanding of the Siebel adapter architecture can help you:

- Understand the relationship between the Siebel adapter and the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
- Understand security boundaries, so that you can improve data security in your solution.
- Understand the Siebel adapter binding properties.
- Troubleshoot installation issues.

In This Section

- [Architecture for BizTalk Adapter for Siebel eBusiness Applications](#)
- [BizTalk Adapter for Siebel eBusiness Applications and the WCF LOB Adapter SDK](#)

Architecture for BizTalk Adapter for Siebel eBusiness Applications

The Microsoft BizTalk Adapter for Siebel eBusiness Applications is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and runs on top of the WCF LOB Adapter SDK run time. The WCF LOB Adapter SDK provides a software framework and tooling infrastructure that the Siebel adapter employs to provide a rich set of features to users and adapter clients.

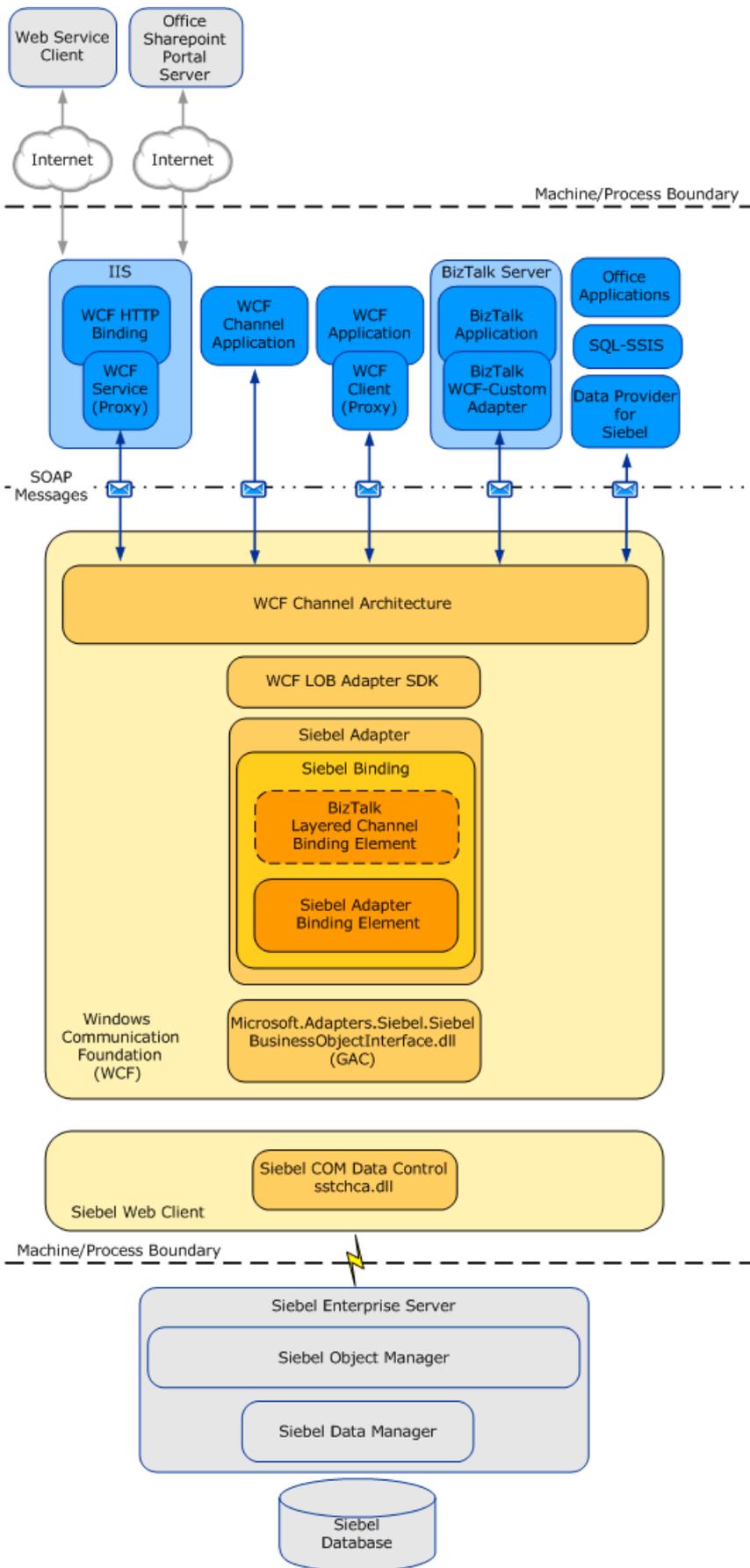
The Siebel adapter is a WCF custom binding. This binding contains a single custom transport binding element that enables communication with a Siebel system. The Siebel adapter is wrapped by the WCF LOB Adapter SDK run time and is exposed to applications through the WCF channel architecture.

The Siebel COM Data Control

The Siebel adapter connects with the Siebel system through the Siebel COM Data Control library (sstchca.dll) and the Microsoft.Adapters.Siebel.SiebelBusinessObjectInterface.dll library. The Siebel COM Data Control is a component of the Siebel Web Client. For information about how to install the Siebel Web Client on your computer, see [Installing the BizTalk Adapter Pack](#).

The Siebel COM Data Control interfaces enable an external client like the Siebel adapter to connect and communicate with a Siebel Application Object Manager on a Siebel Enterprise Server. The Siebel Object Manager and Siebel Enterprise Server as well as other connection parameters are specified in the Siebel adapter connection URI. For more information about the connection URI, see [The Siebel System Connection URI](#).

The following figure shows the end-to-end architecture for solutions that are developed by using the Siebel adapter.



Consuming the Adapter

The Siebel adapter exposes the Siebel system as a WCF service to client applications. To perform operations and access data on the Siebel system, client applications exchange SOAP messages with the Siebel adapter through WCF channels. The preceding figure shows four ways in which the Siebel adapter can be consumed.

- Through a WCF channel model application. A WCF channel model application performs operations on the Siebel system by using the WCF channel model to exchange SOAP messages directly with the Siebel adapter. For more information about developing solutions for the Siebel adapter by using the WCF channel model, see

Developing Applications by Using the WCF Channel Model.

- Through a WCF service model application. A WCF service model application calls methods on a WCF client to perform operations on the Siebel system. A WCF client models the operations exposed by the Siebel adapter as .NET methods. You can use the WCF LOB Adapter SDK or the ServiceModel Metadata Utility Tool (svcutil.exe) to create a WCF client class from metadata exposed by the Siebel adapter. For more information about the WCF service model and the Siebel adapter, see [Developing Applications by Using the WCF Service Model](#).
- Through a BizTalk receive location or send port that is configured to use the Microsoft BizTalk WCF-Custom adapter. The WCF-Custom adapter enables the use of WCF extensibility features. By using the WCF-Custom adapter you can select and configure the Siebel Binding and the behavior for the receive location or send port. BizTalk transactions are supported by the BizTalk Layered Channel Binding Element, which can be loaded by setting a binding property on the Siebel Binding. For more information about how to use the Siebel adapter in BizTalk Server solutions, see [Developing BizTalk Applications](#).
- Through an IIS-hosted Web service. In this scenario, a WCF service proxy generated by using the adapter is hosted in IIS using the standard WCF Http Binding. This exposes the service contract as a Web service to external users. IIS automatically hosts the adapter at run time, which, in turn, communicates with the Siebel system.

The Siebel adapter and the Siebel COM Data Control library are always hosted in-process with the application or service that consumes the adapter.

The Siebel Adapter and WCF

WCF presents a programming model based on the exchange of SOAP messages over channels between clients and services. These messages are sent between endpoints exposed by a communicating client and service. An endpoint consists of:

- An *endpoint address*, which specifies the location at which messages are received.
- A *binding*, which specifies the communication protocols that are used to exchange messages.
- A *contract*, which specifies the operations and data types that are exposed by the endpoint.

A binding consists of one or more binding elements that stack on top of each other to define how messages are exchanged with the endpoint. At a minimum, a binding must specify the transport and encoding that are used to exchange messages with the endpoint. Message exchange between endpoints occurs over a channel stack that is composed of one or more channels. Each channel is a concrete implementation of one of the binding elements in the binding that is configured for the endpoint. For more information about WCF and the WCF programming model, see the WCF Help documentation at <http://go.microsoft.com/fwlink/?LinkId=89726>.

The Microsoft BizTalk Adapter for Siebel eBusiness Applications exposes a WCF custom binding, the Siebel Binding (**Microsoft.Adapters.Siebel.SiebelBinding**). By default, this binding contains a single custom transport binding element, the Siebel Adapter Binding Element (**Microsoft.Adapters.Siebel.SiebelAdapter**), which enables operations on a Siebel system. When using the Siebel adapter with BizTalk Server, you can set the **EnableBizTalkCompatibilityMode** binding property to load a custom binding element—the BizTalk Layered Channel Binding Element—on top of the Siebel Adapter Binding Element. The BizTalk Layered Channel Binding Element is implemented internally by the Siebel adapter and is not exposed outside the Siebel Binding.

Microsoft.Adapters.Siebel.SiebelBinding (the Siebel Binding) and **Microsoft.Adapters.Siebel.SiebelAdapter** (the Siebel Adapter Binding Element) are public classes and are also exposed to the configuration system. Because the Siebel Adapter Binding Element is exposed publicly, you can build your own custom WCF bindings capable of extending the functionality of the Siebel adapter. For example, you could implement a custom binding to support Enterprise Single Sign-on (SSO) in WCF channel or service model programming. The reasons for doing this are to either:

- Aggregate database operations into a single multifunction operation.
- Perform schema transformation between operations that are implemented by a custom application and operations on the Siebel system.

See Also

Other Resources

[Planning and Architecture](#)

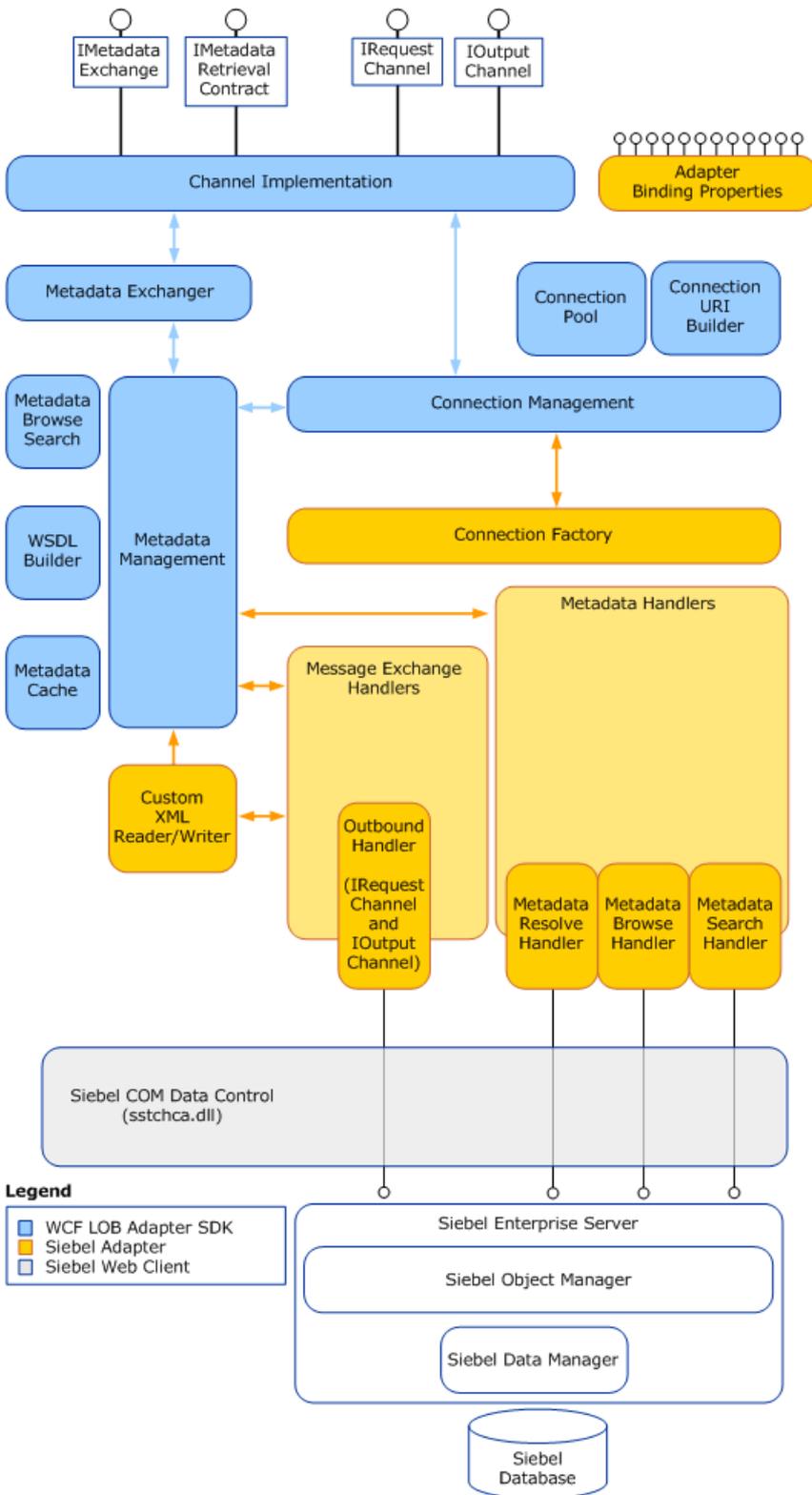
[Understanding BizTalk Adapter for Siebel eBusiness Applications](#)

BizTalk Adapter for Siebel eBusiness Applications and the WCF LOB Adapter SDK

The Microsoft BizTalk Adapter for Siebel eBusiness Applications implements a set of core components that:

- Leverage functionality provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
- Provide connectivity to the Siebel system through the Siebel COM Data Control library (sstchca.dll).

The WCF LOB Adapter SDK is the software layer through which the Siebel adapter interfaces with WCF; Siebel COM Data Control is the layer through which the Siebel adapter interfaces with the Siebel system. The following figure shows the relationships between the internal components of the Siebel adapter and between these components and the Siebel COM Data Control.



See Also

Other Resources

[Planning and Architecture](#)

Security and Protection

The Siebel system often contains sensitive business information such as customer account details. Applications that use the Microsoft BizTalk Adapter for Siebel eBusiness Applications to access and modify this information either locally or across a distributed network might inadvertently expose it to access by unauthorized actors, unless efforts are made to protect and secure the data during transmission. Data protection and security are usually thought of in the following terms:

- *Authorization* controls access to a resource based on the identity of the requester.
- *Authentication* provides mechanisms for verifying the identity of a requester.
- *Data confidentiality* provides mechanisms for protecting the privacy of data through encryption.
- *Data integrity* provides mechanisms to digitally sign data, so that the receiver can ensure that the data has not been altered in-transit.

Another important area of concern is the user-name password credentials that you supply to the Siebel adapter. The adapter uses these credentials to open connections to the Siebel system. These credentials can be supplied in the connection URI; however, because the user name and password are clear text, the Siebel adapter provides alternative methods that you can use to supply these credentials in a more secure manner.

The topics in this section provide guidelines to help you better secure the solutions that you develop with the Siebel adapter.

In This Section

- [Security Considerations Between the Siebel System and the Adapter](#)
- [Security Considerations When Using the Adapter with BizTalk Server](#)
- [Security Considerations When Programming on the Adapter](#)
- [Best Practices](#)

Security Considerations Between the Siebel System and the Adapter

The Siebel adapter can support rsa or mscrypto encryption on the data that it exchanges with the Siebel system. You configure the encryption mode through a query string parameter in the connection URI. For more information about the Siebel connection URI, see [The Siebel System Connection URI](#). For more information about rsa and mscrypto encryption support by Siebel, see the Siebel documentation.

Specifying an encryption mode can help to ensure privacy of data exchanged between the adapter and the Siebel system; however, the Siebel adapter does not provide mechanisms that support authorization, authentication or data integrity on such exchanges. If these issues are a concern in your environment, you must provide a security mechanism to help mitigate them.

One possible mechanism for helping to provide more security across the network is Internet Protocol Security (IPsec). IPsec is a framework of open standards for protecting communications over Internet Protocol (IP) networks. For more information about IPsec and about using IPsec with Microsoft products, see the Microsoft TechNet article "IPsec" at <http://go.microsoft.com/fwlink/?LinkID=196851>.

The Siebel adapter supports authorization and authentication on the connections that it establishes with the Siebel system through user name password credentials that you supply. The Siebel adapter uses these credentials to authenticate the user on the Siebel system when it opens a connection. These credentials provide a level of authorization on the Siebel system for the connection. The Siebel adapter provides a number of methods through which you can supply these credentials. For information about how to more securely provide Siebel credentials in BizTalk solutions, see [Security Considerations When Using the Adapter with BizTalk Server](#). For information about how to more securely provide Siebel system credentials in programming solutions, see [Security Considerations When Programming on the Adapter](#).

Note

The credentials used by the Siebel adapter to establish a connection to the Siebel system do not provide message-level or transport-level authentication or authorization for data traveling across the network. They are only used to open a connection and authenticate the user on the Siebel system.

See Also

Other Resources

[Security and Protection](#)

Security Considerations When Using the Adapter with BizTalk Server

When you configure a send port or a receive port (location) by using the BizTalk Server Administration console, or when you use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas for a BizTalk solution, you must provide credentials for the Siebel system. It is important to supply these credentials in a secure way to help prevent them from being revealed to potentially malicious actors. This topic discusses how to most securely supply credentials for the Microsoft BizTalk Adapter for Siebel eBusiness Applications for BizTalk Server solutions.

A more general discussion of security in the context of BizTalk solutions is an expansive topic and is beyond the scope of this documentation. For information about how you can make your BizTalk solutions more secure, see the "Security and Protection" topic in the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101135>.

How Do I Protect Credentials When I Use the Consume Adapter Service BizTalk Project Add-in?

When you use the Consume Adapter Service Add-in to retrieve message schemas for a BizTalk solution, you must supply a user name and password for the Siebel system. You should only do this from the **Security** tab on the **Configure Adapter** dialog box. This ensures that your credentials will not be displayed in the **Uri** field of the Consume Adapter Service Add-in dialog box, where anyone with access to your computer screen can read them. For more information about how to retrieve message schemas by using the Consume Adapter Service Add-in, including how to enter a user name and password for the Siebel system, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).

How Do I Protect Credentials When I Configure a Send Port or a Receive Location?

BizTalk solutions use the Microsoft BizTalk WCF-Custom adapter to consume WCF services. The Siebel adapter is a WCF custom binding that enables clients to consume the Siebel system as if it were a WCF service. BizTalk solutions consume the Siebel adapter through send ports and receive locations that are configured to use the WCF-Custom adapter, which is, in turn, configured to use the Siebel adapter as its transport. For more information about how to configure send ports and receive ports (receive locations), including how to configure the WCF-Custom adapter, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).

You configure the Siebel system credentials from the **Credentials** tab of the **WCF-Custom Transport Properties** dialog box for send ports or from the **Other** tab of the **WCF-Custom Transport Properties** dialog box for receive locations. Because the WCF-Custom adapter supports Enterprise Single Sign-On (SSO), you can choose to provide either a user name and password or an SSO affiliate application on either of these tabs. The following topics discuss both options.

User Name Password Credentials

You should only supply a user name and password from the **Credentials** tab (for send ports) or the **Other** tab (for receive locations) in the **WCF-Custom Transport Properties** dialog box. This ensures the following:

- Your credentials will not be displayed in the **Uri** field of the dialog box. This prevents those who have access to your screen (or who have permissions that enable them to view the send port or receive location properties) from seeing your credentials.
- Your password will not be written to the binding file if you export the send port or receive port binding. This prevents anyone with access to the file from viewing your password.

Enterprise Single Sign-On and SSO Affiliate Applications

You can configure the WCF-Custom adapter to use Enterprise Single Sign-on (SSO) to get the credentials for the Siebel system. SSO uses a database and a master secret to encrypt and store user credentials. It also provides services to map Microsoft Windows accounts to secondary credentials that are used to access a back-end system. By using SSO, you can map a Windows account to a user name and password on the Siebel system.

SSO uses *affiliate applications* and *SSO mappings* to map credentials to the back-end system. An affiliate application is a logical entity in SSO that refers to a system or an application that requires secondary credentials. An SSO mapping is associated with an affiliate application. It maps a Windows account to the secondary credentials used by that account to access the affiliate system or application. An SSO mapping can be associated with a Windows user account or with a group.

To use SSO with the Siebel adapter, you must do the following.

1. Create an affiliate application in SSO to hold the user name password credentials for the Siebel system. This step is often

performed by someone with special types of SSO administrative privileges.

2. Create a user or group mapping for the affiliate application that maps your Windows account to the user name and password that are used to establish a connection with the Siebel system. Depending on your installation, a user might be able to perform this step or it might require someone with special types of SSO administrative privileges.

Note

When configured for SSO, the WCF-Custom adapter uses services provided by SSO to get the Siebel user name and password from the SSO database. It provides these (unencrypted) to the Siebel adapter, so that the adapter can open a connection to the Siebel system. SSO provides no encryption or protection across the connection between the Siebel adapter and the Siebel system.

For information about how to use SSO, including information about how to create affiliate applications and SSO mappings, see the "Using SSO" topic in the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkId=103052>. For more general information about SSO, see the "Implementing Enterprise Single Sign-On" topic in the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkId=103054>.

The AcceptCredentialsInUri Binding Property

The Siebel adapter surfaces the **AcceptCredentialsInUri** binding property. This property determines whether Siebel system credentials are permitted in the connection URI. By default, **AcceptCredentialsInUri** is **false** and the Siebel adapter throws an exception if credentials are included in the URI.

This property is surfaced because there are certain programming scenarios that require the credentials to be present in the connection URI. This should never be the case when you are configuring a send port or a receive location, or when you are using the Consume Adapter Service Add-in to retrieve message schemas from the Siebel adapter. It is recommended that you do not set **AcceptCredentialsInUri** to **true**. For more information about the Siebel adapter binding properties, see Working with [Working with BizTalk Adapter for Siebel Binding Properties](#).

The **AcceptCredentialsInUri** binding property is not available in BizTalk Server in the **Binding** tab while configuring a WCF-Custom or WCF-Siebel send port. To set the value of the **AcceptCredentialsInUri** binding property, you must open the adapter bindings file (XML file) that is created after you have generated metadata using the Consume Adapter Service Add-in, and then locate this binding property in the file. Specify an appropriate value for this binding property, save the binding file, and then import the binding file in BizTalk Server. See [Importing Bindings](#) for instructions.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Programming on the Adapter

How Do I Protect Credentials When I Use the Add Adapter Service Reference Visual Studio Plug-in?

When you use the Add Adapter Service Reference Plug-in to create a WCF client, you must supply a user name and password for the Siebel system. You should only do this from the **Security** tab on the **Configure Adapter** dialog box. By entering the Siebel credentials from the **Security** tab instead of directly into the **Uri** field, you ensure the following:

- The credentials will not be displayed in the **Uri** field of the Add Adapter Service Reference Plug-in dialog box where anyone with access to your computer screen can read them.
- The credentials will not appear in the configuration file that the Add Adapter Service Reference Plug-in generates.

For more information about how to generate a WCF client by using the Add Adapter Service Reference Plug-in, including how to enter a user name and password for the Siebel system, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).

What Are Best Practices for Setting Credentials in Code?

WCF provides the **ClientCredentials** class to help you configure the credentials that a client communication object, such as a **ChannelFactory**, uses to authenticate itself with a service. By using the **ClientCredentials** class, you ensure that WCF takes whatever authentication mechanisms are specified in that object's channel stack and applies them to the exchange between your client and the service.

Because the Siebel adapter is hosted in-process with its consuming application, it is not imperative to use the **ClientCredentials** class to set credentials on the client communication objects that the consuming application uses. It is, however, considered good practice to do so.

The Siebel adapter encourages the use of the **ClientCredentials** class through the **AcceptCredentialsInUri** binding property. This property specifies whether the adapter will accept the user name and password for the Siebel system in the connection URI. **AcceptCredentialsInUri** defaults to **false**, which means that the adapter will throw an exception if the connection URI contains credentials. You can set **AcceptCredentialsInUri** to **true** to supply credentials in the connection URI. In fact, you must do this in certain cases; for example, when you use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class for Siebel system artifacts.

The following example shows how to use the **Credentials** class to set credentials for the Siebel system on a **ChannelFactory**.

```
// Create binding and endpoint
SiebelBinding binding = new SiebelBinding();
EndpointAddress endpointAddress = new EndpointAddress("siebel://Siebel_server:1234?SiebelObjectManager=obj_mgr&SiebelEnterpriseServer=entserver&Language=enu ");

// Create the channel factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, endpointAddress);

// Set user name and password
factory.Credentials.UserName.UserName = "YourUserName";
factory.Credentials.UserName.Password = "YourPassword";

// Open the channel factory
factory.Open();
```

The following example shows how to use the **ClientCredentials** class to set credentials for the Siebel system on a WCF client.

```
// Initialize a new client for the SQLEXECUTE operation from configuration
BusinessObjects_Account_Account_OperationClient accountAccountClient = new BusinessObjects_Account_Account_OperationClient ("SiebelBinding_BusinessObjects_Account_Account_Operation");

// Set user name and password
accountAccountClient.ClientCredentials.UserName.UserName = "YourUserName";
accountAccountClient.ClientCredentials.UserName.Password = "YourPassword";
```

```
// Open the client
accountAccountClient.Open();
```

How Can I Provide for More Secure Data Exchange Across Process Boundaries?

The Siebel adapter is hosted in-process with the application or service that consumes it. Because the adapter is hosted in-process with the consumer, there is no need to provide security on messages exchanged between the consumer and the Siebel adapter. However, if the consuming application or service sends messages that contain sensitive database information across a process boundary to another service or client, you should take measures to provide adequate protection for this data in your environment. Windows Communication Foundation (WCF) provides many options for helping to secure messages sent between clients and services. For more information about helping to secure messages sent between clients and services in WCF, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>. For more general information about security features that WCF provides, see "Windows Communication Foundation Security" at <http://go.microsoft.com/fwlink/?LinkId=89726>.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Best Practices

This section provides best practices that you should follow to more completely protect sensitive data when you use or develop applications that consume the Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Security Best Practices for the Connection between the Siebel Adapter and the Siebel System

- The Siebel adapter supports mscrypto or rsa encryption on the data exchanged between the adapter and the Siebel system. To help ensure the privacy of the data that is exchanged between the adapter and the Siebel system, you should enable one of these encryption modes.
- The Siebel adapter does not provide mechanisms to help ensure data integrity or to provide for authentication and authorization on the data that is exchanged between it and the Siebel system. You must provide such mechanisms, if these concerns exist in your environment.
- Do not provide user name password credentials for the Siebel system in the connection URI. See the remainder of this topic for alternative methods of providing credentials to the Siebel adapter.

For more information, see [Security Considerations Between the Siebel System and the Adapter](#).

Security Best Practices for Consuming the Siebel Adapter with BizTalk Server

- Do not provide user name password credentials for the Siebel system in the connection URI.
- When you use the Consume Adapter Service Add-in, enter the user name password credentials for the Siebel system from the **Security** tab of the **Configure Adapter** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Siebel adapter on a send port, enter the user name password credentials for the Siebel system from the **Credentials** tab of the **Configure WCF Custom Transport** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the Siebel adapter on a receive location, enter the user name password credentials for the Siebel system from the **Other** tab of the **Configure WCF Custom Transport** dialog box.

For more information, see [Security Considerations When Using the Adapter with BizTalk Server](#).

Security Best Practices for Consuming the Siebel Adapter with Programming Solutions

- It is sometimes necessary to provide the user name password credentials for the Siebel system in the connection URI; however, if possible, you should avoid doing this.
- When you use the Add Adapter Service Reference Visual Studio Plug-in, enter the user name password credentials for the Siebel system from the **Security** tab of the **Configure Adapter** dialog box.
- In WCF Channel Model programming, use the **Credentials** property on the channel factory to set the user name password credentials for the Siebel system.
- In WCF Service Model programming, use the **ClientCredentials** property on the WCF client to set the user name password credentials for the Siebel system.
- If an application that consumes the Siebel adapter sends messages that contain sensitive database information across a process boundary to another service or client, ensure that these messages have sufficient security measures applied to provide adequate data protection in your environment.

For more information see, [Security Considerations When Programming on the Adapter](#).

Security Best Practices for Hosting the Siebel Adapter in IIS

- Hosting the Siebel adapter in Microsoft Internet Information Services (IIS) as a Web service exposes operations surfaced by the Siebel adapter to Web clients. These operations might involve exchanging sensitive data over the Internet, so you should take measures to help ensure that this data is as secure as possible.

WCF provides two standard bindings for HTTP transport: the **BasicHttpBinding** provides basic HTTP transport with no security mechanisms; the **WSHttpBinding** supports both transport-level and message-level security mechanisms.

You can either use the **BasicHttpBinding** over an HTTPS connection, or use the **WSHttpBinding** to help protect your data. The WCF LOB Adapter SDK includes the WCF LOB Adapter Service Development Wizard to generate WCF services for LOB artifacts. This wizard only supports use of **BasicHttpBinding**.

You can also develop a custom HTTP binding to leverage additional security mechanisms that your environment provides. For more information about the security features that WCF provides, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>.

Security Best Practices for WCF Diagnostic Tracing and Message Logging

WCF supports diagnostic tracing and message logging. You configure diagnostic tracing and message logging either through configuration files or by using Windows Management Instrumentation (WMI). Depending on the configuration options you set, WCF diagnostic tracing or message logging can emit sensitive information to log files, where it could potentially be exposed to observation by unauthorized users.

Follow the recommendations provided in the WCF documentation to mitigate potential security threats exposed by enabling these features. At a minimum, you should observe the following best practices for diagnostic tracing and message logging:

- Do not enable "verbose" or "information" tracing in a production environment. This may lead to performance degradation. However, you must enable "warning" and "error" tracing in a production environment. If you enable tracing, you must take proper security measures to protect your data. See the WCF documentation for more information.
- Ensure that log files and configuration files are protected by access control lists (ACLs).

The following warnings apply specifically to the messages that are exchanged between a client application and the Siebel adapter:

- WCF diagnostic tracing can log the header (but not the body) of messages exchanged with the Siebel adapter. Because the message action is in the message header, this reveals the operations invoked on the Siebel adapter by the client.
- If WCF message logging is enabled and **logMessagesAtServiceLevel** is **true**, the message header (but not the message body) of messages exchanged between the adapter client and the Siebel adapter are logged. Because the message action is in the message header, this reveals the operations that the client invoked on the Siebel adapter. If **logEntireMessage** is also **true**, the message body will be logged. This can reveal sensitive database information.

For more information about improving security when you enable diagnostic tracing, see "Security Concerns and Useful Tips for Tracing" at <http://go.microsoft.com/fwlink/?LinkId=89796>. For more information about improving security when you enable message logging, see "Security Concerns for Message Logging" at <http://go.microsoft.com/fwlink/?LinkId=89797>.

See Also

Other Resources

[Security and Protection](#)

Development

The Microsoft BizTalk Adapter for Siebel eBusiness Applications is a Windows Communication Foundation (WCF) custom binding. Client applications can consume the Siebel adapter to invoke operations on Siebel artifacts. The Siebel adapter can be consumed:

- Through a physical port binding in a BizTalk Server solution.
- By invoking methods on an instance of a client proxy.
- As a hosted WCF service.
- By sending SOAP messages over a channel instance in code that uses the WCF channel model.
- Through an ADO.NET interface.

The following table:

- Lists the different operations that can be performed on a Siebel system using the Siebel adapter.
- Indicates which of the approaches (BizTalk Server, WCF service model, WCF channel model, or ADO.NET interface) can be used to perform the operations.
- Provides links to more information about performing the task using the chosen approach.

T a s k	BizTalk Server	WCF Service Model	WCF Channel
B a s i c I n s e r t, U p d a t e, D e l e t e, a n d Q u e r y o p e r a t i o n s o n b u s i n e s s c o m p o n e n t s	Performing Operations on Business Components Using BizTalk Server	Performing Operations on Business Components by Using the WCF Service Model	Performing O...

<p>Performing Operations on Business Components with MVG Fields Using BizTalk Server</p>	<p>Performing Operations on Business Components with MVG Fields by Using the WCF Service Model</p>	
<p>Performing Operations on Business Components with Picklist Fields Using BizTalk Server</p>		
<p>Invoking Business Service Methods Using BizTalk Server</p>		

Invoking Business Service Methods with Integration Objects		
--	--	--

The topics in this section provide information, procedures, and examples to help you develop applications that consume the Siebel adapter in both BizTalk Server and .NET programming solutions. The topics also provide information on other key aspects of using the adapters such as

- [Connecting to the Siebel system.](#)
- [Retrieving metadata from the Siebel system.](#)
- [Using binding properties to configure the adapter.](#)

In This Section

- [Establishing a Connection to the Siebel System](#)
- [Retrieving Metadata for Siebel Operations in Visual Studio](#)
- [Working with BizTalk Adapter for Siebel Binding Properties](#)
- [Developing BizTalk Applications](#)
- [Developing Applications by Using the WCF Service Model](#)
- [Developing Applications by Using the WCF Channel Model](#)
- [Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)
- [Using the Siebel Adapter with Microsoft Office SharePoint Server](#)
- [Samples](#)

Establishing a Connection to the Siebel System

The Microsoft BizTalk Adapter for Siebel eBusiness Applications is a Windows Communication Foundation (WCF) custom binding. As such, it enables communication to a Siebel system through a WCF endpoint address. In WCF the endpoint address identifies the network location of a service and is typically expressed as a Uniform Resource Identifier (URI). The Siebel adapter expresses this location as a connection URI, which contains properties that the Siebel adapter uses to establish a connection to the Siebel system. You must specify a connection URI when you:

- Create a channel factory or a channel listener using the WCF channel model, or create a WCF client or service host using the WCF service model.
- Create a physical port binding in a BizTalk Server solution.
- Use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class, or WCF service interface for a WCF service model solution.
- Use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas from the Siebel adapter for a BizTalk Server solution.
- Use the ServiceModel Metadata Utility tool (svcutil.exe) to generate a WCF client class, or WCF service interface for a WCF service model solution.

The topics in this section describe how to establish a connection between the Microsoft BizTalk Adapter for Siebel eBusiness Applications and the Siebel system by providing you with:

- Information about the connection properties and the structure of the Siebel connection URI.
- Links to topics that show how to establish a connection by using the Siebel adapter.

In This Section

- [The Siebel System Connection URI](#)

See Also

Concepts

[Development](#)

The Siebel System Connection URI

The Microsoft BizTalk Adapter for Siebel eBusiness Applications connection URI contains properties that the adapter uses to establish a connection to the Siebel system.

This topic provides information about the Siebel connection URI and also provides links to other topics that explain how to specify a connection URI in different programming scenarios.

Connection URI for the Siebel Adapter

A typical WCF endpoint address URI is represented as follows:

```
scheme://userinfoparams@hostinfoparams?query_string
```

The endpoint address URI contains the following components:

- scheme is the scheme name.
- userinfoparams is a name-value collection of parameters required for user authentication by the endpoint.
- hostinfoparams is information required to establish the connection to the host; for example, a path.
- query_string is an optional name-value collection of parameters delimited by a question mark (?).

The Siebel connection URI follows this general format and is implemented as follows:

```
siebel://Username=[USER_NAME];Password=[PASSWORD]@[SERVER]:[PORT]?SiebelObjectManager=[SIEBEL_OBJECT_MANAGER_NAME]&SiebelEnterpriseServer=[SERVER_NAME]&Language=[LANGUAGE]&Transport=[TRANSPORT]&Encryption=[ENCRYPTION]&Compression=[COMPRESSION]&SiebelServer=[SIEBEL_SERVER_NAME]&SiebelRepository=[SIEBEL_REPOSITORY_NAME]
```

The following sections describe the properties implemented for each component of the Siebel connection URI.

The Scheme for the Siebel Connection URI

The scheme for the Siebel connection URI is "siebel".

User Information in the Siebel Connection URI

By default, the Siebel adapter throws an exception when Siebel system credentials are specified in the connection URI. This is because these credentials are represented as plain text, which poses an inherent security risk. You can set the **AcceptCredentialsInUri** binding property to control whether the connection URI can contain credentials. If the **AcceptCredentialsInUri** property is **false**, the Siebel adapter throws an exception if the connection URI contains credentials; if the property is **true**, no exception is thrown.

◆ Important

Due to the inherent security risks posed by passing credentials in strings as plain text, it is best not to specify Siebel system credentials in the connection URI.

There are several ways to supply Siebel system credentials without specifying them in the connection URI.

- In code, you can set the **ClientCredentials** property on the appropriate object.
- When you use the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, you can enter the credentials by selecting the **Security** tab of the **Configure Adapter** dialog box.
- When you specify a send port or receive location binding in a BizTalk Server solution, you can enter the credentials by selecting the **Security** tab of the appropriate dialog box.

The user information (userinfoparams) in the Siebel connection URI is represented as a name-value collection of parameters required for user authentication. The following table describes these parameters.

Property	Description
User name	The user name on the Siebel system; this value is case-sensitive. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI.  Note The Siebel adapter preserves the case of the value that you enter for the user name when it opens a connection on the Siebel system.
Password	The password for the user on the Siebel system; this value is case-sensitive. You must set the AcceptCredentialsInUri binding property to true to specify the user name and password in the connection URI.  Note The Siebel adapter preserves the case of the value that you enter for the password when it opens a connection on the Siebel system.

Host Information in the Siebel Connection URI

The Siebel host information (hostinfoparams) specifies the address of the Siebel system in the following format: [SERVER]:[PORT]. Depending on the Siebel server version, the Siebel host information takes different values:

- **For Siebel version 7.5 and earlier**, the host information parameter takes the name of the computer on which Siebel gateway server is installed and the Siebel gateway port number.
- **For Siebel version 7.7 and later**, the host information parameter takes the name of the computer on which the Siebel server is installed and the Siebel connection broker port number.

Important

When you use the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in to connect to a Siebel system, the host information must be provided for the "SiebelGateway" connection property.

Query Information in the Siebel Connection URI

The query information (query_string) in the Siebel connection URI is used to specify additional connection properties.

Property	Description
SiebelObjectManager	The name of the Siebel object manager on the enterprise server. This parameter is required.
SiebelEnterpriseServer	The name of the Siebel Enterprise Server. This parameter is required.
Language	The language of the object manager. This parameter is optional. If it is not specified, the Siebel adapter supplies a default value (enu).
Transport	The transport; only tcpip is supported. This parameter is optional. If it is not specified, the Siebel system supplies a default value (tcpip).
Encryption	The type of encryption to use between the Siebel adapter and the Siebel system. Supported values are none, mscrypto, or rsa. This parameter is optional. If it is not specified, the Siebel system supplies a default value (none).
Compression	The compression algorithm to use between the Siebel adapter and the Siebel system. Supported values are none or zlib. This parameter is optional. If it is not specified, the Siebel system supplies a default value (zlib).

SiebelServer	The Siebel server. Required for all Siebel 7.5 server connections (7.5.2, 7.5.3, etc.); otherwise, do not set this parameter.
SiebelRepository	The Siebel repository. Required if more than one repository exists on the server; otherwise, optional.  Note If more than one repository exists on the server, you must specify a target repository in the SiebelRepository parameter.

For more information about the Siebel parameters that are set in the query information, see your Siebel documentation.

Using Reserved Characters in the Connection URI

The Siebel adapter does not support specifying a connection URI that has special characters for any of the parameter values. If the connection parameter values contain special characters, make sure you do one of the following:

- If you are specifying the URI in Visual Studio using Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. If you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.
- If you are specifying the URI while creating a send or receive port in BizTalk Server Administration console, and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

Using the Connection URI to Connect to the Siebel System

The following is a sample Siebel connection URI.

```
siebel://Username=YourUserName;Password=YourPassword@Siebel_server:1234?SiebelObjectManager=obj_mgr&SiebelEnterpriseServer=entserver&Language=enu
```

Note

This sample URI contains the Siebel system credentials; you must set the **AcceptCredentialsInUri** binding property to **true** to use a connection URI that contains credentials.

For information about how to establish a connection to the Siebel system (including setting connection properties) when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).
- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Configuring a WCF Client for a Siebel System](#).
- Use the WCF ServiceModel Metadata Utility Tool (svcutil.exe), see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Siebel eBusiness Applications](#).

See Also

Concepts

[Development](#)

[Developing Applications by Using the WCF Channel Model](#)

[Developing Applications by Using the WCF Service Model](#)

Other Resources

Retrieving Metadata for Siebel Operations in Visual Studio

The Microsoft BizTalk Adapter for Siebel eBusiness Applications provides two Visual Studio components that you can use to help you develop solutions using the adapter.

- The Consume Adapter Service BizTalk Project Add-in is available in BizTalk Server projects. You use the Consume Adapter Service Add-in to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The Add Adapter Service Reference Visual Studio Plug-in is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

Both of these Visual Studio components simplify development by:

- Providing a Microsoft Windows interface through which you can browse and search for operations that you want to use in your solution.
- Retrieving metadata exposed by the adapter for these target operations.
- Converting that metadata, which is expressed as a Web Services Description Language (WSDL) document by the adapter, into a form that you can use in your solution (XSD message schemas for BizTalk projects or a .NET object representation of a service contract for the WCF service model) and adding it to your project.

This section provides instructions about how to use the Consume Adapter Service Add-in and the Add Adapter Service Reference Plug-in.

In This Section

- [Connecting to the Siebel System in Visual Studio](#)
- [Browsing, Searching, and Retrieving Metadata for Siebel Operations](#)

See Also

Concepts

[Development](#)

Connecting to the Siebel System in Visual Studio

This section provides instructions about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

- The **Consume Adapter Service BizTalk Project Add-in** is available in BizTalk Server projects and is installed as part of the WCF LOB Adapter SDK installation. You use the Consume Adapter Service Add-in to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The **Add Adapter Metadata Wizard** is available in the BizTalk Server projects and is installed as part of the BizTalk Server installation. You use the Add Adapter Metadata Wizard to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).

Note

Because the Siebel adapter is exposed both as a WCF-Custom binding and as BizTalk adapter, you can use either the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard from a BizTalk project to connect to a Siebel system.

- The **Add Adapter Service Reference Visual Studio Plug-in** is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

To use the Consume Adapter Service BizTalk Project Add-in, Add Adapter Metadata Wizard, or the Add Adapter Service Reference Visual Studio Plug-in, you must first connect to the Siebel system. All three methods present a dialog box through which you configure a connection by setting the following:

- **Connection parameters.** These are the parameters that are used to build the connection URI such as the name of the Siebel Enterprise Server.
- **User name password credentials for the Siebel system.** These are used to authenticate you on the Siebel system when the connection is established. You must specify a user name and password.
- **Binding properties.** Binding properties are optional and whether you specify them depends primarily on whether you target operations that require specific binding properties to be set.

At a minimum, when you configure the connection to the Siebel system, you only have to specify binding properties and connection parameters that are needed to establish the connection and that affect the metadata returned by the Siebel adapter for the operations you want to target. However, you might also want to specify values for any additional binding properties and connection parameters that will be used at run time. This is because:

- The Consume Adapter Service Add-in creates a BizTalk port binding file from the binding properties and connection parameters that you specify when you configure the connection and adds this file to your project.
- The Add Adapter Service Reference Plug-in creates an app.config file from the binding properties and connection properties that you specify when you configure the connection and adds this file in your project directory.

In This Section

- [Connecting to the Siebel System in Visual Studio Using Consume Adapter Service Add-in](#)
- [Connecting to the Siebel System in Visual Studio Using Add Adapter Metadata Wizard](#)

- [Connecting to the Siebel System in Visual Studio Using Add Adapter Service Reference Plug-in](#)

See Also

Concepts

[Retrieving Metadata for Siebel Operations in Visual Studio](#)

Connecting to the Siebel System in Visual Studio Using Consume Adapter Service Add-in

The Consume Adapter Service BizTalk Project Add-in is installed when you install WCF LOB Adapter SDK. The Consume Adapter Service Add-in loads all the WCF-Custom bindings installed on the computer. To connect to a Siebel system using the WCF-based Siebel adapter in a BizTalk project, you must use the **siebelBinding**.

Connecting to a Siebel System Using Consume Adapter Service Add-in

Perform the following steps to connect to a Siebel system using the Consume Adapter Service Add-in.

To connect to a Siebel system

1. To connect using the Consume Adapter Service Add-in in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

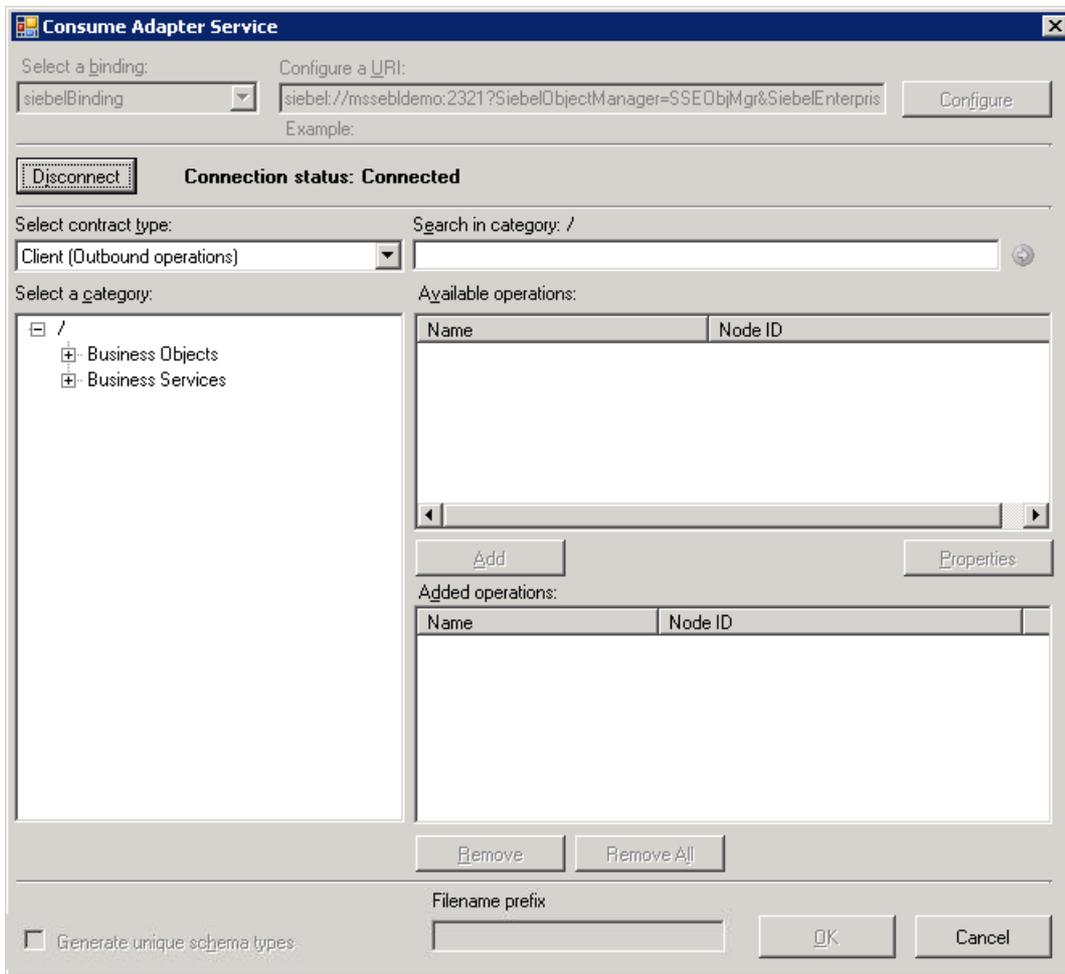
- d. Click **Add**. The Consume Adapter Service Add-in opens.
2. From the **Select a binding** drop-down list, select **siebelBinding**, and then click **Configure**.
 3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username**.
 4. Specify the user name and password to connect to the Siebel system.
 5. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).

Note

If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

6. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
7. Click **OK**.
8. Click **Connect**. Once the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on the Siebel system. For example, the **Business Objects** node contains all the business objects and business components that can be invoked on the Siebel system. Similarly, the **Business Services** node contains all the business services in the Siebel system you connected to. For more information about these nodes, see [Metadata Node IDs](#).

Connecting to the Siebel System in Visual Studio Using Add Adapter Metadata Wizard

The Siebel adapter is also exposed as a BizTalk adapter and hence, you can use the Add Adapter Metadata Wizard to generate schema for the operations you want to perform on the Siebel system using the adapter.

Connecting to a Siebel System Using Add Adapter Metadata Wizard

Perform the following steps to connect to a Siebel system using the Add Adapter Metadata Wizard.

To connect to a Siebel system

1. To connect using the Add Adapter Metadata Wizard in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

- d. Click **Add**. The Add Adapter Metadata Wizard opens.
- e. In the Add Adapter Metadata Wizard, select WCF-Siebel. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-Siebel port configured in BizTalk, select the port from the Port list.

- f. Click **Next**.
2. From the **Select a binding** drop-down list, select **siebelBinding**, and then click **Configure**.
 3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username**.
 4. Specify the user name and password to connect to the Siebel system.
 5. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).

📌 Note
If the connection parameters contain any reserved characters, you must specify them as-is in the URI Properties tab, that is, without using any escape characters. However, if you specify the URI directly in the Configure a URI field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

6. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

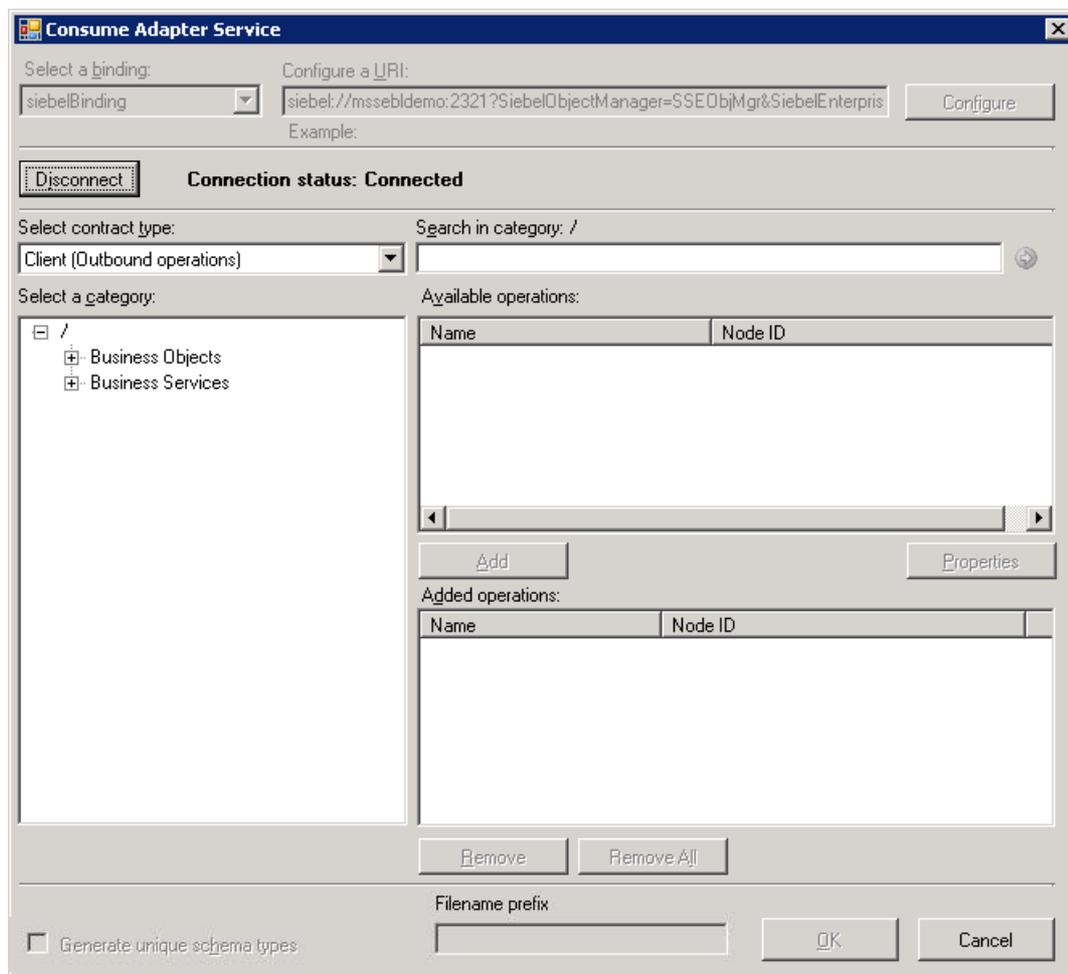
📌 Note

If you are generating metadata using Add Adapter Metadata Wizard and you selected an existing WCF-Siebel send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

7. Click **OK**.

8. Click **Connect**. Once the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on the Siebel system. For example, the **Business Objects** node contains all the business objects and business components that can be invoked on the Siebel system. Similarly, the **Business Services** node contains all the business services in the Siebel system you connected to. For more information about these nodes, see [Metadata Node IDs](#).

Connecting to the Siebel System in Visual Studio Using Add Adapter Service Reference Plug-in

To connect to a Siebel system using the Siebel adapter in a .NET programming solution, you must use the Add Adapter Service Reference Visual Studio Plug-in. This topic provides instructions on how to use the Add Adapter Service Reference Plug-in.

Connecting to a Siebel System Using Add Adapter Service Reference Plug-in

Perform the following steps to connect to a Siebel system using the Add Adapter Service Reference Plug-in.

To connect to a Siebel system

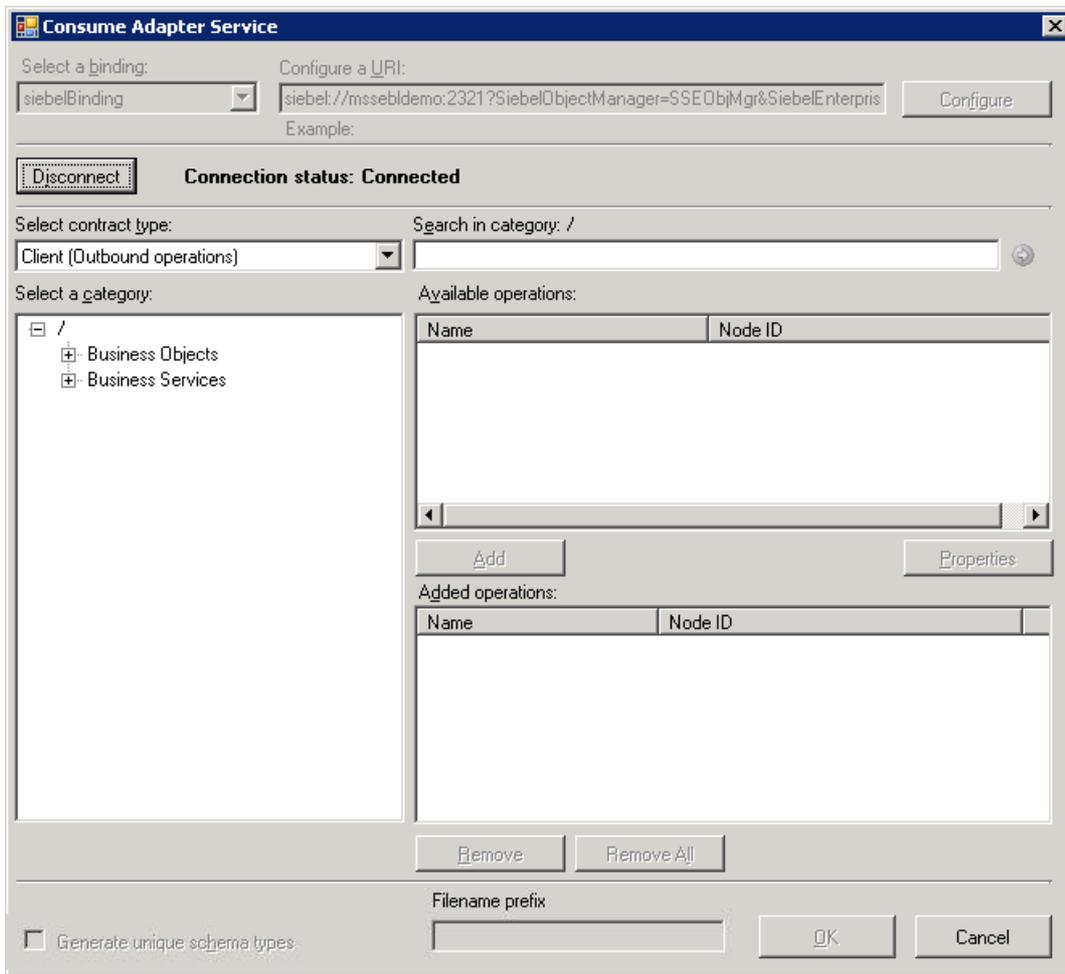
1. To connect using the Consume Adapter Service Add-in in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project in Solution Explorer, and then click **Add Adapter Service Reference**. The Add Adapter Service Reference Plug-in opens.
2. From the **Select a binding** drop-down list, select **siebelBinding**, and then click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, select **Username**.
4. Specify the user name and password to connect to the Siebel system.
5. Click the **URI Properties** tab, and specify values for the connection parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).

Note

If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

6. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
7. Click **OK**.
8. Click **Connect**. Once the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established. The graphical user interface is same for the Add Adapter Service Reference Plug-in.



The Add Adapter Service Reference Plug-in displays different nodes containing various operations that can be performed on the Siebel system. For example, the **Business Objects** node contains all the business objects and business components that can be invoked on the Siebel system. Similarly, the **Business Services** node contains all the business services in the Siebel system you connected to. For more information about these nodes, see [Metadata Node IDs](#).

Browsing, Searching, and Retrieving Metadata for Siebel Operations

This section provides information about how to use the Consume Adapter Service BizTalk Project Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Visual Studio Plug-in. By using these Visual Studio components, you can:

- Browse for operations for which to retrieve metadata.
- Search for operations for which to retrieve metadata.
- Add message schemas for selected operations and port binding configuration files to a BizTalk Server project when using the Consume Adapter Service Add-in.
- Add a WCF client class or a WCF service contract (interface) for selected operations and a configuration file (app.config) to a non-BizTalk programming project when using the Add Adapter Service Reference Plug-in.

Note

The Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so all three components are covered in the same topics.

Prerequisites

You must connect to a Siebel system before you can browse, search, or retrieve metadata for target operations. For information about how to connect to a Siebel system when you use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in, see [Connecting to the Siebel System in Visual Studio](#).

Browsing Metadata

While browsing metadata using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in, the Siebel adapter surfaces:

- Operations that can be performed on Siebel business components such as Insert, Query, Update, and Delete.
- Business service methods that can be invoked by the adapter clients.

Note

By using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in, you can browse category and operation nodes using a Windows interface.

For more information about browsing Siebel metadata, see [How Does the Adapter Surface Siebel Metadata?](#)

Perform the following steps to browse metadata of a Siebel system using the Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in.

To browse metadata in a Siebel system

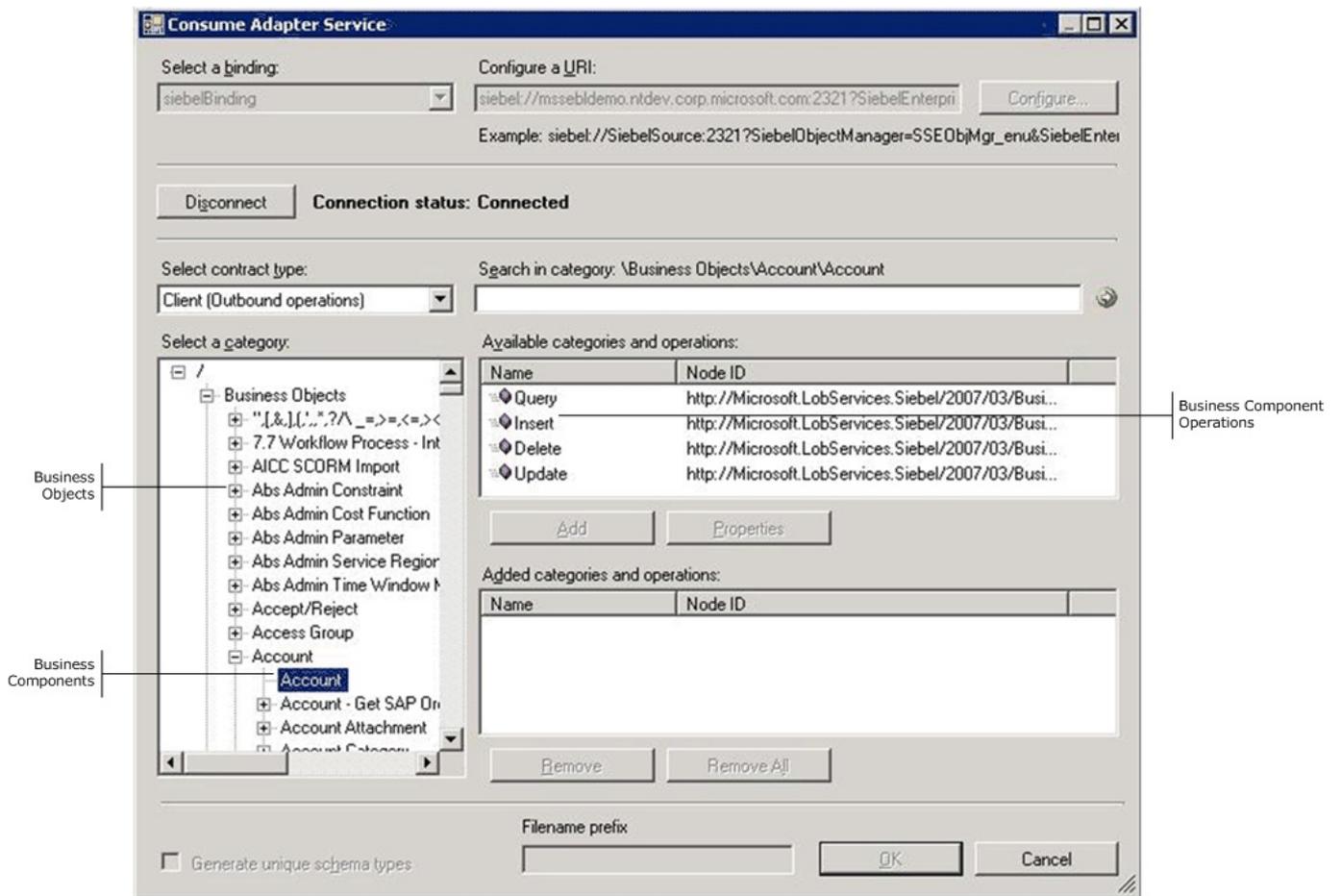
1. Connect to a Siebel system using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in. See [Connecting to the Siebel System in Visual Studio](#) for instructions.
2. From the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound or outbound operations using the adapter.
3. The **Select a category** box lists the **Business Objects** and **Business Services** nodes. Click the **Business Objects** node to see a list of business objects in the **Available categories and operations** box. Alternatively, you can see the list of business objects by expanding the **Business Objects** node.

Tip

You can directly go to the "immediate" category node or subcategory nodes in the tree, by typing the name of the artifact in while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Account** business object, keep the focus on the **Business Objects** node, and then type **Account**.

4. Click the business objects to see the list of business components for a particular business object. Alternatively, you can see the list of business components by expanding a business object.
5. Click the business components to see the list of operations supported for the particular business component.

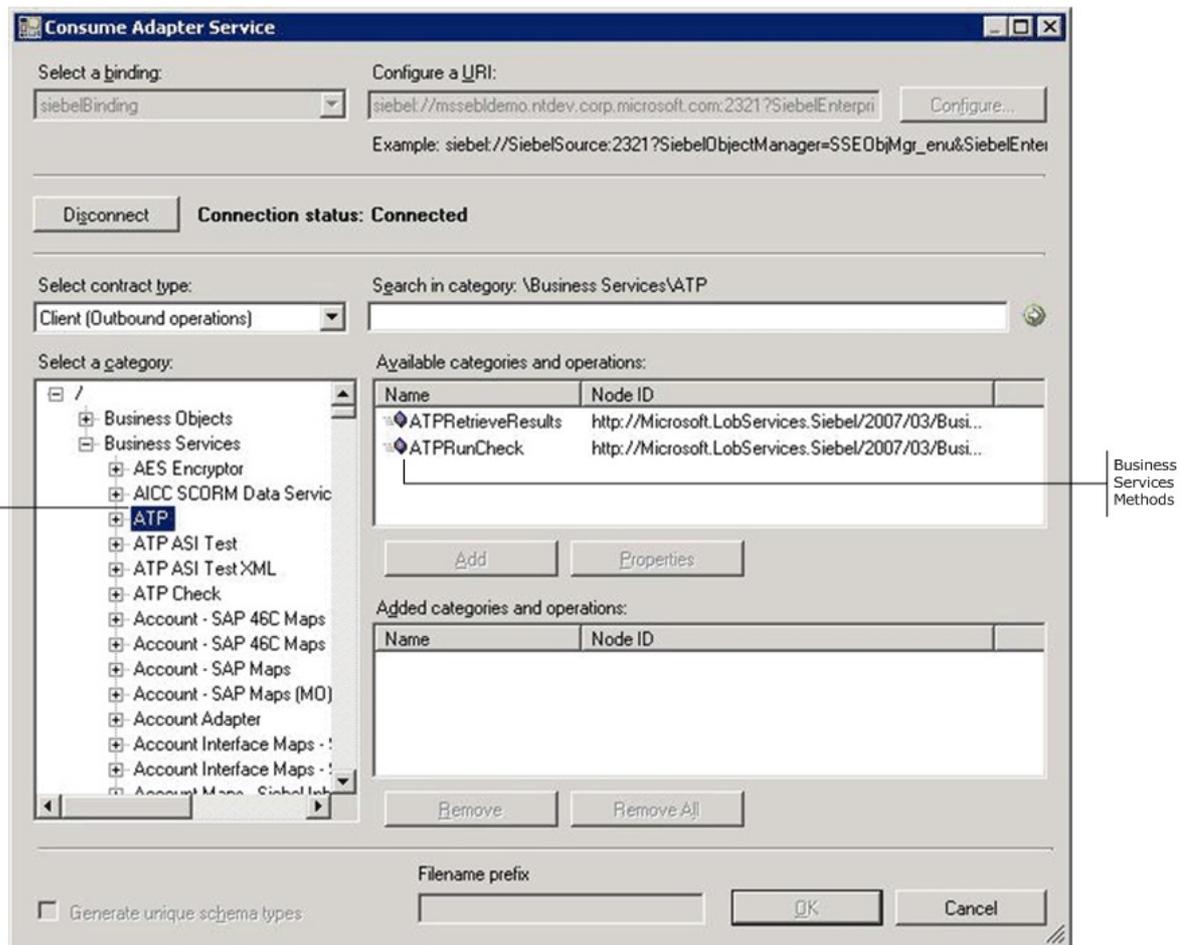
The following figure shows the Consume Adapter Service Add-in, which lists the business object, the business components, and the supported operations.



6. Click the **Business Services** node to see a list of business services in the **Available categories and operations** box. Alternatively, you can see the list of tables by expanding the **Business Services** node.

7. Click the business services to see the list of corresponding business service methods.

The following figure shows the Consume Adapter Service Add-in, which lists the business services and the corresponding business service methods.



Searching Metadata

While searching Siebel metadata using Add Adapter Service Reference Plug-in, Add Adapter Metadata Wizard, or Consume Adapter Service Add-in, the Siebel adapter:

- Supports wildcard and escape characters.
- Enables search immediately under the node at which the search operation is performed. For example, to search for a business service, you must be searching under \Business Services.

The following table lists the special characters that can be used for search and their interpretation by the Siebel adapter.

Special character	Interpretation
? (question mark)	Matches exactly one character For example, A? matches AB, AC, AD.
* (asterisk)	Matches zero or more characters. For example, A* matches A, AB, ABC.

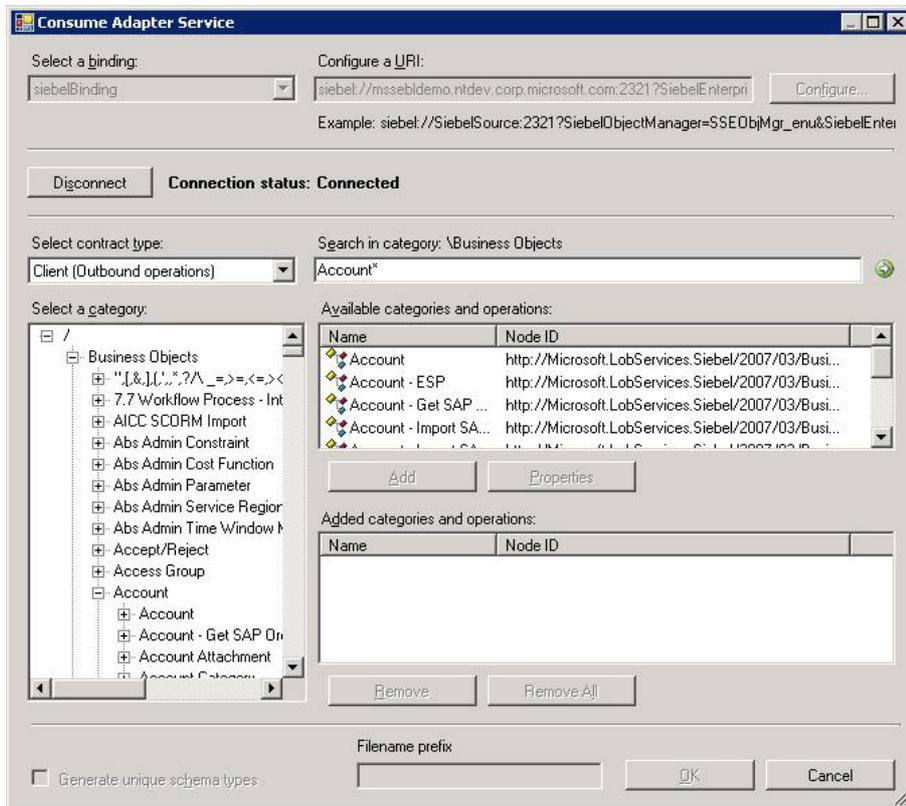
Perform the following steps to search metadata in a Siebel system using the Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in.

To search metadata in a Siebel system

1. Connect to a Siebel system using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or Add Adapter Service Reference Plug-in. See [Connecting to the Siebel System in Visual Studio](#) for instructions.
2. From the **Select contract type** drop-down list, select the **Client (Outbound operations)** contract.
3. In the **Select a category** box, click the **Business Object** node.
4. To search for a particular business object, click the **Business Objects** node in the **Search in category** text box and enter a search expression. For example, to search for business objects that have names beginning with "Account", type **Account*** in the text box.
5. Click the button with the right-arrow icon to start the search. After the search is complete, the **Available categories and operations** box lists the business objects that satisfy the search criteria.
6. To search for a particular business component under a business object, click a business object and in the **Search in category** text box enter a search expression. For example, to search for business components that have names beginning with "Account", type **Account*** in the text box.

7. Click the button with the right-arrow icon to start the search. After the search is complete, the **Available categories and operations** box lists the business components that satisfy the search criteria.
8. To search particular operations for a business component, click a business component and in the **Search in category** text box enter a search expression. For example, to search for operations having names starting with "Query", type **Query*** in the text box.
9. Click the button with the right-arrow icon to start the search. After the search is complete, the **Available categories and operations** box lists the business components that satisfy the search criteria.

The following figure shows the Consume Adapter Service Add-in, which lists the search result.



Perform a similar procedure to search under the **Business Service** node.

Generating Schema for BizTalk projects

You can use the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard to generate schema for selected Siebel artifacts. Once you have browsed and searched for the artifacts you want to invoke, you can generate schema for those artifacts and send messages, conforming to the schema, to Siebel.

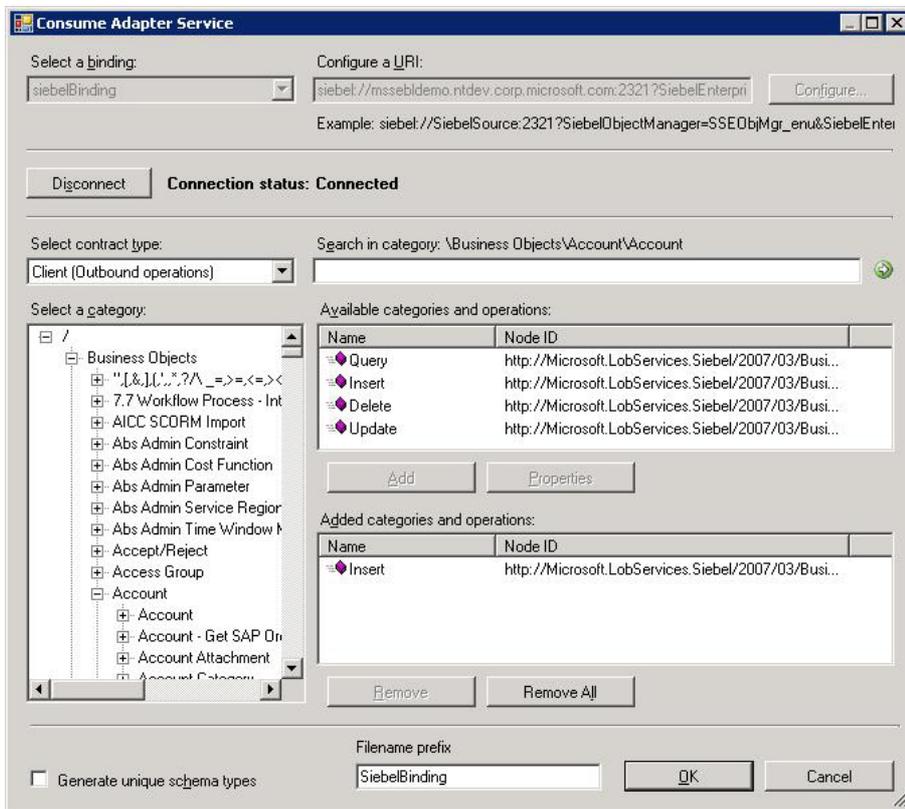
Note

You can select category nodes to return all the operations in that category's sub-tree—for example, you can select a business component (to generate schema for all the operations in the business component) or a select specific operations on a business component (for example, Insert and Delete) to generate schema for only those operations. For more information about the nodes, see [Metadata Node IDs](#).

To generate schema for Siebel artifacts

1. Connect to a Siebel system using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard. See [Connecting to the Siebel System in Visual Studio](#) for instructions.
2. From the **Select contract type** drop-down list, select the **Client (Outbound operations)** contract.
3. In the **Select a category** box, expand the business object or business service node.
4. In the **Available categories and operations** box, select the business components or business services or the corresponding operations for which you want to generate metadata, and then click **Add**. The selected functional areas or operations are listed in the **Added categories and operations** box.

The following figure shows the Consume Adapter Service Add-in, which lists the selected operations.



If you want to generate schema for multiple operations, there may be some duplicate element definitions among these schema that may cause failure in compiling the BizTalk project. For example, consider a scenario where you generate schema for an operation "Op1". The schema for "Op1" contains a parameter of complex data type "CT1". After generating the schema for "Op1" you close the Consume Adapter Service Add-in and re-open it to generate schema for another operation "Op2". Assume that "Op2" also contains a parameter of complex data type "CT1". After you exit the Consume Adapter Service Add-in and compile the project, you will get compilation errors because the complex data type "CT1" is defined twice in different XSD files. In such situations, we recommend the following:

- Generate schema for all the operations in a single run of Consume Adapter Service Add-in. This ensures that the Consume Adapter Service Add-in generates only one definition for the complex data type "CT1".
- If you want to generate schema for multiple operations across different runs of Consume Adapter Service Add-in, make sure you select the **Generate unique schema types** check box so that the generated XSD files contain unique namespaces for the complex data type "CT1".

5. Click **OK**. The schema file is saved with an .xsd extension at the same location as the BizTalk project.

By default, the files are created with the naming convention "SiebelBindingSchema<n>.xsd", where 'n' can be 1, 2, and so on, depending on the number of schema files created. Alternatively, you can provide a custom name to the schema files by entering a name in the **Filename prefix** text box. The Consume Adapter Service Add-in now creates schema files with the naming convention <file name prefix>Schema<n>.xsd.

Note

The Consume Adapter Service Add-in also creates a binding file (an XML file) containing the binding properties that you specified when generating the schema for an operation and the SOAP action to invoke the operation. You can import this binding file in the BizTalk Server Administration console to create a WCF-Custom port with the connection URI, binding properties, and the SOAP action set. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

6. On the **File** menu, click **Save All**.

Generating a WCF Client Using the Add Adapter Service Reference Plug-in

You can use the Add Adapter Service Reference Plug-in to generate WCF client code for outbound operations on a Siebel system.

To generate a WCF client

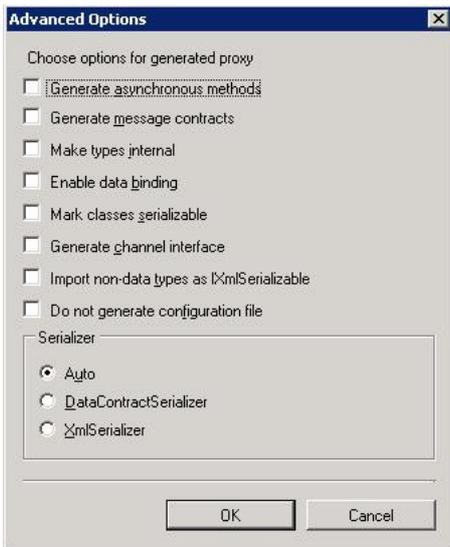
1. In the Add Adapter Service Reference Plug-in, from the **Select contract type** drop-down list, select **Client (Outbound operations)**.
2. In the **Select a category** box, expand the business object or business service node.
3. In the **Available categories and operations** box, select the business components, business services, or the corresponding operations for which you want to generate a WCF client, and then click **Add**. The selected functional areas or operations are listed in the **Added categories and operations** box. You can select any node that is listed in the **Available categories and operations** box. If you select a category node, then all of the operations available under that node and its sub-nodes will be selected.

Important

Depending on the categories and operations that you select, more than one WCF client class may be generated. For more information, see [Generating a WCF Client for Siebel Artifacts](#).

4. For most scenarios the default serialization options are sufficient; however, if needed, you can control several aspects about the code that is generated and the type of serializer that is used. To set these options:
 - a. To open the **Advanced Options** box, click **Advanced Options**.
 - b. In the **Advanced Options** box under **Choose options for generated proxy**, select the options that you want. For example, you can select whether asynchronous methods are generated for the WCF client, or you can disable the generation of a configuration file.
 - c. Under **Serializer**, select the serializer that should be used.

The following figure shows the **Advanced Options** box with the default selections (**Auto** is selected for the serializer and no other options are selected).



The options that you can configure in the **Advanced Options** box are equivalent to some of the options available when you use the ServiceModel Metadata Utility Tool (svcutil.exe). For more information about these options, see "ServiceModel Metadata Utility Tool (Svcutil.exe)" at <http://go.microsoft.com/fwlink/?LinkId=99821>.

5. Click **OK**. The Add Adapter Service Reference Plug-in saves the WCF client class and helper code for the operations and categories that you have selected in your project directory. By default, a configuration file is also saved. For more information, see [Generating a WCF Client for Siebel Artifacts](#).

See Also

Concepts

[Retrieving Metadata for Siebel Operations in Visual Studio](#)

Working with BizTalk Adapter for Siebel Binding Properties

The Microsoft BizTalk Adapter for Siebel eBusiness Applications surfaces several binding properties that enable you to control some of its run-time and design-time behavior. This section describes these binding properties and provides links to topics that explain how you can set them.

The Siebel Adapter Binding Properties

The following table shows the Siebel adapter binding properties grouped by category. The category refers to the node under which each binding property appears in the dialog boxes that are presented by different applications to configure the adapter (or binding).

Binding Property	Category	Description	.NET Type
EnableBizTalkCompatibilityMode	General	Specifies whether the BizTalk Layered Channel Binding Element should be loaded. Set this to True to load the binding element. Otherwise, set this to False . When using the adapters from BizTalk Server, you must always set the property to True . When using the adapters from Visual Studio, you must always set the property to False .	bool (System.Boolean)
Name	General	Specifies the name of the file generated by the Add Adapter Service Reference Visual Studio Plug-in to hold the WCF client class. The Add Adapter Service Reference Plug-in forms the file name by appending "Client" to the value of the Name property. The default is "SiebelBinding"; for this value, the generated file will be named "SiebelBindingClient".	string
CloseTimeout	General	Specifies the WCF connection close timeout. The default is 1 minute.	System.DateTime
OpenTimeout	General	Specifies the WCF connection open timeout. The default is 1 minute.	System.DateTime
ReceiveTimeout	General	Specifies the WCF message receive timeout. The default is 10 minutes.	System.DateTime
SendTimeout	General	Specifies the WCF message send timeout. The default is 1 minute.	System.DateTime
EnableConnectionPooling	Connection	Specifies whether the Siebel adapter connection pool is enabled. The default is true , which specifies that the connection pool is enabled.	bool (System.Boolean)

IdleConnectionTimeout	Connection	Specifies the Siebel adapter idle connection timeout. When a connection is idle for a period that exceeds this timeout, the connection will be disposed. The default is 1 minute.	System.Data.Time
MaxConnectionsPerSystem	Connection	Specifies the maximum number of connections in the Siebel adapter connection pool. The default is 5. MaxConnectionsPerSystem is a static property within an application domain. This means that when you change MaxConnectionsPerSystem for one binding instance in an application domain, the new value applies to all objects created from all binding instances within that application domain.	int (System.Int32)
EnablePerformanceCounters	Diagnostics	Specifies whether the WCF LOB Adapter SDK performance counters and the Siebel adapter LOB Latency performance counter are enabled. The default is true ; performance counters are enabled. The Siebel adapter LOB Latency performance counter measures the total time the adapter spends in making calls to the Siebel system.	boolean (System.Boolean)
LogData	Diagnostics	Specifies whether to capture business data in traces. The default is false ; business data is not captured.	boolean (System.Boolean)
AcceptCredentialsInUri	Not surfaced by the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in.	Specifies whether the Siebel connection URI can contain user credentials for the Siebel system. The default is false , which disables user credentials in the connection URI. If AcceptCredentialsInUri is false and the connection URI contains user credentials, the Siebel adapter throws an exception. You can set AcceptCredentialsInUri to true if you must specify credentials in the URI. For more information, see The Siebel System Connection URI .	boolean (System.Boolean)

How Do I Set Siebel Binding Properties?

You can set the Siebel binding properties when you configure a connection to a Siebel system. For information about how to set binding properties when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to the Siebel System in Visual Studio](#).

◆ Important

While using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file (an XML file) or the app.config file respectively. You must manually add the binding property and its value in the binding file or the app.config file, if required.

- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Configuring a WCF Client for a Siebel System](#).
- Use the WCF ServiceModel Metadata Utility Tool (svcutil.exe), see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Siebel eBusiness Applications](#).

See Also
Concepts
[Development](#)

Developing BizTalk Applications

Developing BizTalk applications involves creating a BizTalk project in Visual Studio and using the Consume Adapter Service BizTalk Project Add-in to generate XML schema. Once you have generated the schema, you can either use Content-Based Routing (CBR) or create BizTalk orchestrations to send and receive messages that conform to the generated schema.

CBR can be used in scenarios where the messages being sent to a Siebel system do not require any intensive processing. For example, if you know that the receive port will be receiving messages only of a certain type, you can add a filter to the send port to route the messages matching the filter expression to the send port.

In BizTalk orchestrations, you create send and receive ports to send and receive messages to and from the WCF-Custom adapter, which in turn sends the messages to BizTalk Server. This section provides information about using BizTalk orchestrations to perform operations on a Siebel system using the Siebel adapter. The Siebel adapter in turn uses the WCF-Custom adapter that can interact with a WCF binding.

◆ Important

To use the Siebel adapter with Microsoft BizTalk Server, you must always set the **EnableBizTalkCompatibilityMode** binding property to **True**. For instructions about how to set the binding properties, see [Specifying Binding Properties](#).

◆ Important

Unlike the earlier version of the adapters shipped with BizTalk Server, the Siebel adapter shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console. This is because the latest Siebel adapter is a WCF custom binding. So, while the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF custom bindings and hence, does not display the WCF-based Siebel adapter.

Also, to generate metadata you must use the Consume Adapter Service BizTalk Project Add-in and not the Add Adapter Metadata Wizard, as used for the previous version of the adapter. For instructions on using the Consume Adapter Service Add-in, see [Retrieving Metadata for Siebel Operations in Visual Studio](#). For more such differences between the adapter versions, see [Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#).

In This Section

- [Before You Develop BizTalk Applications](#)
- [Building Blocks for Developing BizTalk Applications](#)
- [Performing Operations on Business Components Using BizTalk Server](#)
- [Performing Operations on Business Components with MVG Fields Using BizTalk Server](#)
- [Performing Operations on Business Components with Picklist Fields Using BizTalk Server](#)
- [Invoking Business Service Methods Using BizTalk Server](#)
- [Invoking Business Service Methods with Integration Objects](#)

See Also

Concepts

[Development](#)

Before You Develop BizTalk Applications

This section provides information about certain prerequisites tasks that you must perform before developing BizTalk applications using the Siebel adapter. The section also lists some BizTalk Server tools that will be used to develop BizTalk applications.

In This Section

- [Create a Strong-Name Key File](#)
- [Learn About BizTalk Server Tools](#)

Create a Strong-Name Key File

You must create a strong name key file to build projects in Microsoft Visual Studio. A strong name consists of the project's identity — its simple text name, version number, and culture information (if provided) — plus a public key and a digital signature. The strong name key file contains the public key and the private key.

Important

You must create one strong name key file, one time before developing a BizTalk application.

To create a strong name key file

1. Click **Start**, point to **All Programs**, point to **Microsoft Visual Studio 2010**, point to **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
2. At the command prompt navigate to the location where you want to create the key file. For example, type **cd C:\Sample**, and then press ENTER.
3. At the command prompt, type **sn -k <key file name>.snk**, and then press ENTER.

Note

You should receive a message at the command prompt stating that the key pair was written to the strong-name key file.

4. At the command prompt, type **exit**, and then press ENTER.

See Also

Concepts

[Learn About BizTalk Server Tools](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Learn About BizTalk Server Tools

The topics on how to use the Siebel adapter in [Developing BizTalk Applications](#) assume that you have working knowledge of a number of BizTalk Server tools. You will use the following tools to develop BizTalk applications using Siebel adapter:

- Microsoft Visual Studio 2010.
- Orchestration Designer
- Pipeline Designer
- BizTalk Mapper
- BizTalk Server Administration console

Prerequisites

You must install BizTalk Server before you can access the BizTalk Server tools.

BizTalk Server Tools

The following table provides you with a list of top-level topics that explain how to use each of the listed tools.

Tool	Topics in BizTalk Server Documentation
Visual Studio	<ul style="list-style-type: none"> • Using Visual Studio (http://go.microsoft.com/fwlink/?LinkId=107180) • Working with BizTalk Projects (http://go.microsoft.com/fwlink/?LinkId=107181) • Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=107182) <p>Learn more about Visual Studio:</p> <ul style="list-style-type: none"> • Introduction to Solutions, Projects, and Items (http://go.microsoft.com/fwlink/?LinkId=58503) • Solution Explorer Concepts (http://go.microsoft.com/fwlink/?LinkId=58504)
Orchestration Designer	<ul style="list-style-type: none"> • Creating Orchestrations Using Orchestration Designer (http://go.microsoft.com/fwlink/?LinkId=107184)
Pipeline Designer	<ul style="list-style-type: none"> • Creating Pipelines Using Pipeline Designer (http://go.microsoft.com/fwlink/?LinkId=107185)
BizTalk Mapper	<ul style="list-style-type: none"> • Creating Maps Using BizTalk Mapper (http://go.microsoft.com/fwlink/?LinkId=107186)
BizTalk Server Administration console	<ul style="list-style-type: none"> • Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkId=107187)

See Also

Tasks

[Create a Strong-Name Key File](#)

Other Resources

Building Blocks for Developing BizTalk Applications

Performing operations on a Siebel system using Siebel adapter involves two set of activities: design-time activities and run-time activities. To perform operations on a Siebel system by using the Siebel adapter with BizTalk Server, you must perform a set of design-time and run-time tasks using Visual Studio and BizTalk Server Administration console respectively. This section provides an overview of these tasks. All the topics in this section, which demonstrate how to perform specific operations on a Siebel system using BizTalk Server, are modeled on these high-level tasks.

Design-time Tasks

The WCF LOB Adapter SDK provides capability to browse, search, and retrieve the Siebel metadata for business components and business services in the form of XML Schema definition languages (XSDs) using the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard. The XSDs are specific to the operation you wish to perform on the Siebel system and the Consume Adapter Service Add-in or Add Adapter Metadata Wizard is available only when you create a BizTalk Project. At design time you may need to perform the following tasks.

- Create BizTalk project and generate schema.** To start with, you must create a BizTalk project in Microsoft Visual Studio and generate the schema for the business components or the business services you will invoke in the Siebel system. For example, if you want to insert a record into the Account business component, you must generate the metadata for the Insert operation for the Account business component. In this step you use the Consume Adapter Service Add-in or Add Adapter Metadata Wizard to generate the schema. For more information, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).
- Set up an orchestration.** Once you have generated the schema, you must set up an orchestration by using the Orchestration Designer. For a basic orchestration, you add the Send and Receive shapes along with the Send and Receive logical ports. In the later steps, you will map these logical ports to physical ports by using the BizTalk Server Administration console. The orchestration uses these ports to pick up messages sent by an adapter client. The orchestration then passes the messages to the Siebel system. Once a response is received from the Siebel system, the orchestration passes the response to the adapter client.
- Create messages and link to schema.** In your orchestration, you must create messages that will be mapped to the schema you generated in the first step. Typically, you would create a request and a response message. These messages are mapped to corresponding request and response schemas.
- Map message shapes to messages and ports.** In your orchestration, you must now map each shape you added in the second step to messages you created in the third step. You must also map a message shape to the port on which that message will be sent.

For example, if the first shape in your orchestration is a Receive shape that will receive a message, you will map this shape to a "request" message and the port that sends the request message.

- Build and deploy the BizTalk project.** After you have set up the orchestration and mapped messages, ports, and schemas you must build the BizTalk solution. For building a project in Visual Studio, you will need an assembly key file. After you successfully build the solution, you must deploy the solution.

Note

More detailed description of these high level tasks, including procedural information is provided under the topics that follow.

Once the solution is deployed, your design time tasks are accomplished. You must now perform the run time tasks.

Run-time Tasks

- Configure the application.** The BizTalk project you deployed at design time will show up in the BizTalk Server Administration console as an orchestration. You must configure this orchestration by mapping the logical ports you created at design time to physical ports that you must now create using the BizTalk Server Administration console.

On the physical ports, you must specify an "action" or "action mapping". This action corresponds to the operation you

want to perform on the Siebel system. You need to set the action if you are not using dynamic actions. For more information about dynamic actions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

- **Start the application.** After the application is configured, you must start the application, and drop input messages at a defined file location. The orchestration consumes the input messages and passes them to the Siebel system and receives a response. This response will be available to you at another defined file location.

To accomplish these high-level design-time and run-time tasks, you must also perform other tasks. For example, when you use the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard to generate the schema, you must specify a connection URI to connect to the Siebel system. This section provides information on such repetitive tasks that you must perform as you develop BizTalk applications by using the Siebel adapter.

In This Section

- [Adding the Siebel Adapter to BizTalk Server Administration Console](#)
- [Specifying the Connection URI](#)
- [Specifying Credentials for the Siebel System](#)
- [Specifying Binding Properties](#)
- [Specifying SOAP Action](#)
- [Manually Configuring a Physical Port Binding to the Siebel Adapter](#)
- [Configuring a Physical Port Binding Using a Port Binding File](#)
- [Configuring Dynamic Ports](#)
- [Reusing Adapter Bindings](#)

Adding the Siebel Adapter to BizTalk Server Administration Console

The Siebel adapter can be used in BizTalk either as a WCF-Custom port or a WCF-Siebel port. If you want to use the Siebel adapter through a WCF-Custom port, you do not need to add the WCF-Custom port to the BizTalk Server Administration console because the WCF-Custom port is added to the BizTalk Server Administration console by default. However, if you want to use the Siebel adapter through a WCF-Siebel port, you must first add the WCF-Siebel adapter to the BizTalk Server Administration console.

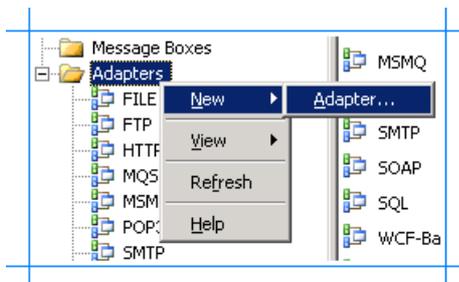
This topic provides instructions on how to add the WCF-Siebel adapter to the BizTalk Server Administration console.

◆ Important

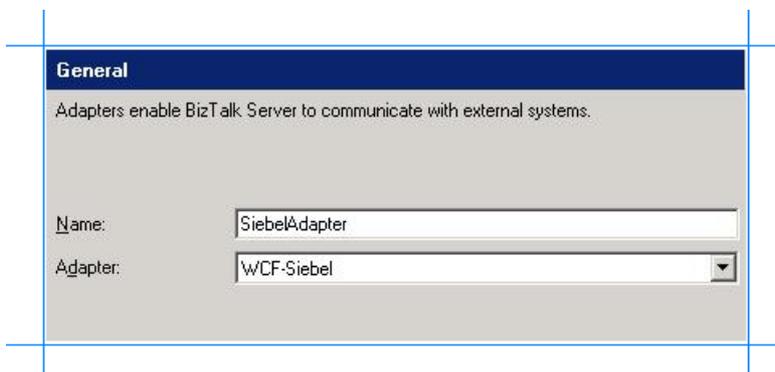
You need not perform these tasks if you want to configure a WCF-Custom port for the Siebel adapter.

To add the Siebel Adapter

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, expand **Platform Settings**, and then click **Adapters**.
3. Right-click **Adapters**, point to **New**, and click **Adapter**.



4. In the **Adapter Properties** dialog box, specify a name for the adapter and from the **Adapter** list, select **WCF-Siebel**.



5. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying the Connection URI

A connection URI is a connection string to connect to a Siebel system. The connection URI contains various parameters required to establish connection with a Siebel system. You must specify this connection URI both at the design time and the run time. At design time, you must specify the URI to connect to the Siebel system to generate the metadata. At run time, you must specify the URI to connect to the Siebel system to perform operations.

Specifying Connection URI at Design Time

For design time, you must specify the connection URI using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify connection URI using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **siebelBinding**, and click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab. From the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Siebel system.
6. In the **Configure Adapter** dialog box, click the **URI Properties** tab and specify values for different parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).
7. In the **Configure Adapter** dialog box, click the **Binding Properties** tab and specify the binding values, if any, which are required to be specified before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
8. Click **OK**.

To specify connection URI using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Wizard, select **WCF-Siebel**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-Siebel port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **siebelBinding**, and click

Configure.

7. In the **Configure Adapter** dialog box, click the **Security** tab. From the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Siebel system.
8. In the **Configure Adapter** dialog box, click the **URI Properties** tab and specify values for different parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).
9. In the **Configure Adapter** dialog box, click the **Binding Properties** tab and specify the binding values, if any, which are required to be specified before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

Note

If you selected an existing WCF-Siebel send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

10. Click **OK**.

Specifying Connection URI at Run Time

For run time, you must specify the URI as part of the WCF-Custom or WCF-Siebel port configuration in the BizTalk Server Administration Console.

To specify connection URI for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Address (URI)** text box, specify the connection URI to connect to the Siebel system. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).
6. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **siebelBinding**.
7. To create a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and do one of the following:
 - a. Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - b. Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.
8. Click **OK**.

To specify connection URI for the WCF-Siebel port

1. Start the BizTalk Server Administration console.
2. Add the WCF-Siebel adapter to the BizTalk Server Administration console. For instructions, see [Adding the Siebel Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-Siebel adapter you added earlier, and

then click **Configure**.

5. In the transport properties dialog box, click the **General** tab.
6. Click the **Configure** button and provide values for the connection parameters. For more information about the connection URI for the Siebel adapter, see [The Siebel System Connection URI](#).
7. In the transport properties dialog box, click the **Binding** tab and specify values for binding properties.
8. To create a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and do one of the following:
 - a. Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - b. Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.
9. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying Credentials for the Siebel System

The Siebel adapter requires the adapter clients to provide client credentials. The adapter uses these credentials to authenticate the user with the Siebel system and to establish a connection.

Adapter clients can provide the client credentials either at design time or run time. At design time, credentials are required to generate the metadata. At run time, credentials are required to perform operations on the Siebel system. This section provides information about specifying client credentials at design time and run time.

Specifying Client Credentials at Design Time

For design time, you must specify the credentials using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify client credentials using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **siebelBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Siebel system.
6. Click **OK**.

To specify client credentials using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Wizard, select **WCF-Siebel**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-Siebel port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **siebelBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab and from the **Client credential type** drop-down list box, select **Username** and specify the user name and password to connect to the Siebel system.

8. Click **OK**.

Specifying Client Credentials at Run Time

For run time, you must specify the client credentials as part of the WCF-Custom or WCF-Siebel port configuration in the BizTalk Server Administration console.

To specify credentials for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. For a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.
5. Click **OK**.

Note

BizTalk Adapter Pack also supports the Enterprise Single Sign-On (ESSO) system. ESSO is only applicable in the BizTalk scenario, in which the WCF adapter is aware of ESSO affiliate applications. For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

To specify credentials for the WCF-Siebel port

1. Start the BizTalk Server Administration console.
2. Add the WCF-Siebel adapter to the BizTalk Server Administration console. For instructions, see [Adding the Siebel Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-Siebel port you added earlier, and then click **Configure**.
5. For a send port, in the transport properties dialog box, click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.
6. Click **OK**.

Note

BizTalk Adapter Pack also supports the Enterprise Single Sign-On (ESSO) system. ESSO is only applicable in the BizTalk scenario, in which the WCF adapter is aware of ESSO affiliate applications. For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

See Also
Tasks

Specifying Binding Properties

The Siebel adapter surfaces several binding properties that enable you to control some of its behavioral characteristics. This section provides information about setting the binding properties from Visual Studio (design time) and from the BizTalk Server Administration Console (run time). At design time, you must specify the binding properties to generate schema for specific operations. At run time, you must specify the binding properties as part of the send port for sending messages to the Siebel system.

For information about the binding properties, including a list of binding properties for Siebel adapter, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

Specifying Binding Properties at Design Time

For design time, you must specify the binding properties from the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard dialog box.

To specify binding properties using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **siebelBinding**, and click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Binding Properties** tab and specify the binding values, if any, which are required to be specified before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
6. Click **OK**.

To specify binding properties using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Wizard, select **WCF-Siebel**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-Siebel port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list select **siebelBinding**, and click **Configure**.

7. In the **Configure Adapter** dialog box, click the **Binding Properties** tab and specify the binding values, if any, which are required to be specified before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

 **Note**

If you selected an existing WCF-Siebel send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

8. Click **OK**.

Specifying Binding Properties at Run Time

For run time, you must specify the binding properties as part of the WCF-Custom or the WCF-Siebel port configuration in the BizTalk Server Administration console.

To specify binding properties for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the **Port Properties** dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab.
5. From the **Binding Type** drop-down list, select **siebelBinding**.
6. In the **Configuration** box, specify the values for the different binding properties and click **OK**.

To specify binding properties for the WCF-Siebel port

1. Start the BizTalk Server Administration console.
2. Add the WCF-Siebel adapter to the BizTalk Server Administration console. For instructions, see [Adding the Siebel Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the **Port Properties** dialog box, from the **Type** drop-down list, select the WCF-Siebel port you added earlier, and then click **Configure**.
5. In the transport properties dialog box, click the **Binding** tab and specify values for the binding properties.
6. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying SOAP Action

To perform any operation on the Siebel system using the WCF-based Siebel adapter, adapter users must specify a SOAP action. The SOAP action communicates to the adapter what action should be performed. You can specify the SOAP action either at design time or at run time. However, if you specify the SOAP action both at design time and run time, the action you specified at design time will be overridden.

For more information about specifying SOAP action, see <http://go.microsoft.com/fwlink/?LinkId=102063>.

Specifying SOAP Action at Design Time

For design time, you must specify the SOAP action as part of orchestration by including an expression shape.

To specify SOAP action at design time

1. In the BizTalk orchestration include an Expression shape by dragging it from the **BizTalk Orchestration** toolbox.
2. Double-click the **Expression** shape to open the BizTalk Expression Editor.
3. Specify the action in the BizTalk Expression Editor. For example:

```
OutboundMessage(WCF.Action)="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert"
```

For more information about **Expression** shape and the BizTalk Expression Editor, see <http://go.microsoft.com/fwlink/?LinkId=102064>.

Specifying SOAP Action at Run Time

For run time, you must specify the SOAP action as part of the WCF-Custom or WCF-Siebel port properties dialog box.

To specify a SOAP action for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:

- **By using the single action format.** Use this format if the WCF-Custom port sends and receive messages for a single operation. For example:

```
http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert
```

- **By using the action mapping format.** Use this format if a single WCF-Custom port sends and receives messages for more than one operation. For example, if a single WCF-Custom port sends and receives messages for Op1 (to perform an Insert operation on Account business component) and Op2 (to perform an Update operation on Account business component), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert " />
  <Operation Name="Op2" Action="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Update " />
</BtsActionMapping>
```

This approach provides greater flexibility in terms of specifying a set of actions and hence enabling messages

belonging to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about action format for each operation, see individual topics under [Technical Reference](#).

To specify a SOAP action for the WCF-Siebel port

1. Start the BizTalk Server Administration console.
2. Add the WCF-Siebel adapter to the BizTalk Server Administration console. For instructions, see [Adding the Siebel Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-Siebel adapter you add earlier, and then click **Configure**.
5. In the port properties dialog box, click the **General** tab.
6. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:
 - **By using the single action format.** Use this format if the WCF-Custom port sends and receive messages for a single operation. For example:

```
http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert
```

- **By using the action mapping format.** Use this format if a single WCF-Custom port sends and receives messages for more than one operation. For example, if a single WCF-Custom port sends and receives messages for Op1 (to perform an Insert operation on Account business component) and Op2 (to perform an Update operation on Account business component), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>  
  <Operation Name="Op1" Action="http://Microsoft.LobServices.Siebel/2007/03/Busine  
ssObjects/Account/Account/Insert " />  
  <Operation Name="Op2" Action="http://Microsoft.LobServices.Siebel/2007/03/Busine  
ssObjects/Account/Account/Update " />  
</BtsActionMapping>
```

This approach provides greater flexibility in terms of specifying a set of actions and hence enabling messages belonging to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about action format for each operation, see individual topics under [Technical Reference](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Manually Configuring a Physical Port Binding to the Siebel Adapter

This section provides information about configuring the Microsoft BizTalk Adapter for Siebel eBusiness Applications as a WCF custom binding by using the BizTalk Server Administration console. After deploying the adapter, you will be able to send and receive messages from the Siebel system by using the BizTalk Server Administration Console. The steps for deploying the adapter vary depending on the direction of communication between BizTalk Server and Siebel adapter. You may choose to configure a Send or a Send-Receive port. Your choices are summarized in the following table:

Port direction	Communication pattern	Direction of communication to choose from
Send	One-way	I will always be sending messages on this port.
Send-Receive	Request-response	I will be sending a request and receiving a response.

For more information, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Note

Receive ports are not supported because the Siebel adapter does not enable inbound operations from the Siebel system.

Note

You can also configure the WCF-Custom send ports by importing a binding configuration file that is created by the Consume Adapter Service Add-in as part of metadata generation. For instructions on configuring ports using this binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

In This Section

- [Configuring a Port Using the WCF-Custom Adapter](#)
- [Configuring a Port Using the WCF-Siebel Adapter](#)

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring a Port Using the WCF-Custom Adapter

This topic provides instructions on how to configure WCF-Custom send ports to perform outbound operations on a Siebel system using the Siebel adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to a Siebel System

Perform the following steps to configure a WCF-Custom send port for sending messages to a Siebel system using the BizTalk Server Administration console.

To deploy adapters for sending messages to a Siebel system

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you wish to deploy the Siebel adapter.
4. Right-click **Send Ports**, point to **New**, and point to a type of port you want to configure depending on the mode of communication between BizTalk Server and the Siebel system.
5. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
6. From the **Type** drop-down list, select **WCF-Custom** and click **Configure**.
7. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab and in the **Address (URI)** field specify the connection URI to connect to a Siebel system. For more information about the connection URI, see [The Siebel System Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on the Account business component is:

```
http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert
```
 - c. Click the **Binding** tab and from the **Binding Type** drop-down list, select **siebelBinding**. You can also specify the different binding properties exposed by the Siebel adapter. For more information, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
 - d. Click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.
8. From the **Send handler** drop-down list, select **BizTalkServerApplication**.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

e. To return to the **Send Port Properties** dialog box, click **OK**.

9. If you chose Static One-Way Send Port in step 4, specify a send pipeline. From the **Send pipeline** drop-down list, select the pipeline corresponding to XMLTransmit.
10. If you chose Static Solicit-Response Port in step 4, specify send and receive pipelines.
 - a. From the **Send pipeline** drop-down list, select the pipeline corresponding to XMLTransmit.
 - b. From the **Receive pipeline** drop-down list, select the pipeline corresponding to XMLReceive.
11. Click **OK**.

See Also

Other Resources

[Manually Configuring a Physical Port Binding to the Siebel Adapter](#)

Configuring a Port Using the WCF-Siebel Adapter

This topic provides instructions on how to configure WCF-Siebel send ports to perform outbound operations on a Siebel system using the Siebel adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to a Siebel System

Perform the following steps to configure a WCF-Siebel send port for sending messages to a Siebel system using the BizTalk Server Administration console.

To deploy adapters for sending messages to a Siebel system

1. Start the BizTalk Server Administration console.
2. Add the WCF-Siebel adapter to the BizTalk Server Administration console. For instructions, see [Adding the Siebel Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
4. Expand the application under which you wish to deploy the Siebel adapter.
5. Right-click **Send Ports**, point to **New**, and point to a type of port you want to configure depending on the mode of communication between BizTalk Server and the Siebel system.
6. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
7. From the **Type** drop-down list, select the WCF-Siebel adapter you added earlier and click **Configure**.
8. In the transport properties dialog box, do the following:
 - a. Click the **General** tab, click the **Configure** button, and provide values for the connection parameters. For more information about the connection URI, see [The Siebel System Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on the Account business component is:

```
http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert
```

- c. Click the **Binding** tab and specify values for the binding properties. For more information, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
- d. Click the **Credentials** tab and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to a Siebel system.
 - Select the **Use Single Sign-On** option, and specify an affiliate ESSO application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

- e. To return to the **Send Port Properties** dialog box, click **OK**.

9. From the **Send handler** drop-down list, select **BizTalkServerApplication**.
10. If you chose Static One-Way Send Port in step 5, specify a send pipeline. From the **Send pipeline** drop-down list, select the pipeline corresponding to XMLTransmit.
11. If you chose Static Solicit-Response Port in step 5, specify send and receive pipelines.
 - a. From the **Send pipeline** drop-down list, select the pipeline corresponding to XMLTransmit.
 - b. From the **Receive pipeline** drop-down list, select the pipeline corresponding to XMLReceive.
12. Click **OK**.

See Also

Other Resources

[Manually Configuring a Physical Port Binding to the Siebel Adapter](#)

Configuring a Physical Port Binding Using a Port Binding File

When you use the Consume Adapter Service BizTalk Project Add-in to generate metadata for a Siebel artifact, other than the schema files, the Consume Adapter Service Add-in also generates a port binding file. You can import this binding file into your BizTalk application to create a physical send port. For instructions on importing binding files, see [Importing Bindings](#). If you import this binding file, you do not have to manually create a physical send port.

◆ Important

While using the Consume Adapter Service Add-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file. You must manually add the binding property and its value in the binding file, if required.

Creating a port using the port binding file always creates a two-way send port. If you want to create one-way port, you can create it manually by following the procedure mentioned in [Manually Configuring a Physical Port Binding to the Siebel Adapter](#). Or, you can follow the workarounds documented in this topic to modify the port binding file to create one-way ports.

◆ Important

Using the Add Adapter Metadata Wizard does not create a port binding file using which you can create a WCF-Siebel port. However, you could make some changes to the port binding file generated by the Consume Adapter Service Add-in and use it to create a WCF-Siebel port. For more information, see [Configuring a WCF-Siebel Port Using the Port Binding File Generated Using Consume Adapter Service Add-in](#).

Following are some key points that you must understand with respect to the binding file generated by the Consume Adapter Service Add-in:

- The files are created with a specific naming convention. If you generated metadata for outbound operations, that is to send messages to the Siebel system, the name of the file is `WcfSendPort_SiebelBinding_Custom.bindinginfo.xml`.
- The file contains information about the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. When you import this binding file to create a port, all the relevant information required to configure a physical port is automatically set on the port.

◆ Important

By default, the action on the send port is mapped to the operation name for which you generate metadata. For example, if you generate metadata for Insert operation on the Account Business component, the action on the port is set to `<Operation Name="Insert" Action="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert " />`. However, the operation name on the logical send port you create in the BizTalk orchestration might not be the same. You must ensure that the operation name in the logical port (in the BizTalk orchestration) and the physical send port (in BizTalk Server Administration console) are the same. If not, you will receive an error while sending messages to the Siebel system through the send port.

- You only need to provide the credentials for the port to connect to the Siebel system. While the binding file does retain the username with which to connect, for security reasons the binding file does not contain the password.

Key Considerations for Using the Port Binding File

- When you import the binding file, you might get a dialog message informing that the BizTalk application name in the binding file does not match the application name to which you are importing the binding file. You can safely ignore this message and continue.
- The binding file also contains names of ports and receive locations. If the BizTalk application to which you are importing the binding file creates a port or a receive location that has the same name as an already existing port in the same BizTalk application, you will get an error. You must manually edit the bindings file to specify a unique name for the ports or receive locations.

- By default, the port binding file always contains definitions for two-way send ports. When you import this file in a BizTalk application, it creates a two-way send port. However, you may have an orchestration that has a one-way send port. So, when you configure such an orchestration and use the port created by importing the binding file, the port is not available in the list. This happens because the logical port you created as part of the orchestration is a one-way port while the physical port created in the orchestration is a two-way port. In such cases, you can edit the bindings file to make the following changes:

For this	Do this
<p>To edit the port bindings file to configure a one-way send port</p>	<ol style="list-style-type: none"> In the following excerpt, change the value of IsTwoWay property to false. Originally, this is set to true. <div data-bbox="695 533 1495 651" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><SendPort Name="port_name" IsStatic="true" IsTwoWay="false" BindingOption="0"></pre> </div> Comment out the following excerpts: <div data-bbox="695 741 1495 1070" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipeline Name="Microsoft.BizTalk.DefaultPipelines.XMLReceive" FullyQualifiedName="Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=token" Type="1" TrackingOption="None" Description="" /></pre> </div> <div data-bbox="695 1093 1495 1178" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipelineData xsi:nil="true" /></pre> </div>

Configuring a WCF-Siebel Port Using the Port Binding File Generated Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in also creates a port binding file that you can import in BizTalk Server Administration console. You can use the same port binding file to also create the BizTalk WCF-Siebel port in BizTalk Server Administration console. However, before creating WCF-Siebel port you must perform the following tasks to modify the port binding file.

1. Open the port binding file in a text editor.
2. Search and replace "WCF-Custom" with the name with which you added the WCF-Siebel adapter in BizTalk Server Administration console. For example, if you added the WCF-Siebel adapter as "SiebelAdapter", replace "WCF-Custom" with "SiebelAdapter".
3. Search for the "ConfigurationClsid" attribute, and replace the existing value of the attribute with "7971A78D-AE8F-42B4-834D-3A957FD945E9".
4. Save and close the binding file.
5. Import the binding file in BizTalk Server Administration console. For instructions on how to import the binding file, see [Importing Bindings](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring Dynamic Ports

In BizTalk Server, you can configure dynamic ports for a WCF-Custom adapter. Because Siebel adapter is a WCF-based adapter, you can dynamically configure a port for the Siebel adapter by using message context properties.

For the Siebel adapter, the URI, action, and binding might be determined from a property on an incoming message, and then specified in the **Expression** shape, as shown in the following example:

```
Request2=Request1;
Request2(WCF.Action)="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert";
Request2(WCF.BindingType)="siebelBinding";
Request2(WCF.UserName)="YourUserName";
Request2(WCF.Password)="YourPassword";
SendPort(Microsoft.XLANGs.BaseTypes.Address)="siebel://mySiebelServer:1234?SiebelObjectManager=SSEObjMgr&SiebelEnterpriseServer=myEnterpriseServer&Language=enu";
SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="WCF-Custom";
```

Note

If you are using a WCF-Siebel adapter in BizTalk Server Administration console, you can also specify the transport type as `SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="SiebelAdapter"`, where **SiebelAdapter** is the name with which you added the WCF-Siebel adapter in BizTalk Server Administration console.

In the preceding example:

- Request2 message is being created from Request1 message. Both the messages map to an operation schema, which is generated using the Consume Adapter Service BizTalk Project Add-in.
- SendPort is the name of the logical send port in the BizTalk orchestration.

The Expression shape is part of the BizTalk orchestration. When you deploy the orchestration, the WCF-Custom send port is also created.

For more information on configuring dynamic ports, see <http://go.microsoft.com/fwlink/?LinkId=106735>.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Reusing Adapter Bindings

A binding creates a mapping between a logical endpoint, such as an orchestration port or a role link, and a physical endpoint, such as a send and receive port or party. This enables communication between different components of a BizTalk business solution. You can create bindings by using the BizTalk Server Administration console.

What Is a Binding File?

A binding file is an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. The binding file describes what host each orchestration is bound to and its trust level as well as the settings for each send port, receive port, receive location, and party that has been configured. You can generate binding files and then apply the bindings they contain to an assembly, application, or group to avoid needing to manually configure bindings in different deployment environments.

For more information about bindings and binding files, see the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

A binding file is not automatically generated for a BizTalk assembly, application, or group, but you can generate a binding file by exporting bindings, as described in [Exporting Bindings](#). You can then import the binding file into an application or group, as described in [Importing Bindings](#).

In This Section

- [Exporting Bindings](#)
- [Importing Bindings](#)

Exporting Bindings

This section describes how to export bindings for a BizTalk application into an XML file. You can then import the bindings from the XML file into another BizTalk application. Importing bindings overwrites any existing bindings of the same name in the application. You can also add bindings to an application, which does not overwrite existing bindings. The bindings that you add do not take effect until you import the application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To export bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application whose bindings you want to export, point to **Export**, and then click **Bindings**.
4. On the Export Bindings page, in **Export to file**, type the absolute path of the .xml file to which to export the bindings.

Example: **C:\Bindings\Application1Bindings.Binding1.xml**

5. Ensure that **Export all bindings from the current application** is selected.
6. To export all party information for the group, select the **Export Global Party information** check box.
7. Click **OK**.

The bindings are exported into an XML file in the location that you specified.

See Also

Tasks

[Importing Bindings](#)

Concepts

[Reusing Adapter Bindings](#)

Importing Bindings

This topic describes how to use the BizTalk Server Administration console to import bindings into a BizTalk application from an XML file.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To import bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application into which you want to import bindings, point to **Import**, and then click **Bindings**.
4. Click the binding file and click **Open**.

The artifacts in the binding file are written to the application. They display in appropriate folders of the application

See Also

Tasks

[Exporting Bindings](#)

Concepts

[Reusing Adapter Bindings](#)

Performing Operations on Business Components Using BizTalk Server

The Siebel adapter surfaces operations that can be invoked on a business component. The operations on business components can be categorized as:

- Standard operations on business components. These include Insert, Query, Update, and Delete. This topic discusses how to perform these operations using BizTalk Server.
- Operations on business components containing multivalued group fields. These include the standard operations as well as Associate, Disassociate, Query_[MVG_Child_Business_Comp]. For more information about performing these operations, see [Performing Operations on Business Components with MVG Fields Using BizTalk Server](#)
- Operations on business components containing picklist fields. For more information about performing these operations, see [Performing Operations on Business Components with Picklist Fields Using BizTalk Server](#).

For more information about how the Siebel adapter supports operations on business components, see [Operations on Business Components](#). For more information about the structure of SOAP messages for performing these operations, see [Message Schemas for Business Component Operations](#).

How to Perform Operations on a Business Component

Performing an operation on a Siebel system using the Siebel adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform an operation on a business component, these tasks are:

1. Create a BizTalk project and generate schema for operation you want to invoke on the business component.
2. Create messages in the BizTalk project for sending and receiving messages from the Siebel system.
3. Create an orchestration to invoke an operation in the Siebel system.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, SiebelAccount, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to invoke operations on a business component, schema are generated for the Insert operation in the Account business component. See [Retrieving Metadata for Siebel Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Siebel system and the other to receive a response.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the orchestration view the BizTalk project, if not already open. Click **View**, point to **Other Windows**, and click **Orchestration View**.
2. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
3. Right-click the newly create message and select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>SiebelAccount.SiebelBindingSchema.Insert</i> , where <i>SiebelAccount</i> is the name of your BizTalk project. <i>SiebelBindingSchema</i> is the schema generated for invoking the <i>Insert</i> operation on <i>Account</i> business component.

5. Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

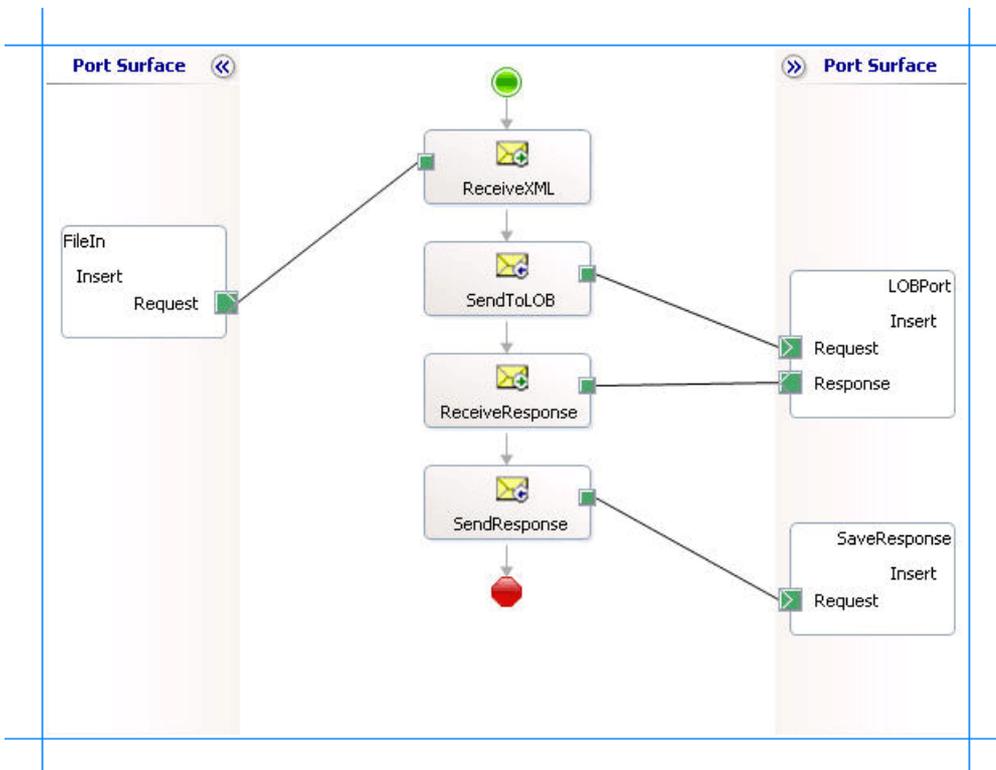
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and select <i>SiebelAccount.SiebelBindingSchema.InsertResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on a Siebel business component. In this orchestration, you drop a request message at a defined receive location. The Siebel adapter consumes this message and passes it on to the Siebel system. The response from the Siebel system is saved to another location. A typical orchestration for performing operations on Siebel business components would contain:

- Send and receive shapes to send messages to Siebel and receive responses.
- A one-way receive port to receive request messages to send to Siebel.
- A two-way send port to send request messages to Siebel and receive responses.
- A one-way send port to send the responses from Siebel to a folder.

A sample orchestration for the *Insert* operation on an *Account* business component resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Shape Type	Properties
ReceiveXML	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveXML</i> Set Activate to <i>True</i>
SendToLOB	Send	<ul style="list-style-type: none"> Set Name to <i>SendToLOB</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the *Port* column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> Set Identifier to <i>FileIn</i> Set Type to <i>FileInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values to be set to specify messages for action shapes and linking them to the ports. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Properties
ReceiveXML	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Insert.Request</i>
SendToLOB	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Insert.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Insert.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Insert.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=102360>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Siebel system.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Siebel system.
- Define a physical WCF-Custom or WCF-Siebel send port to send messages to the Siebel system. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Administration Console to create send ports (for outbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for performing an *Insert* operation on the *Account* business component in Siebel. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-Siebel send port to send messages to the Siebel system is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop the request message to the FILE receive location. The schema of the request message must conform to the schema of the Insert operation (on Account business component) you generated earlier. For example, the request message to insert a record in the Account business component is:

```
<Insert xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Operation">
  <ArrayOfAccountInsertRecord>
    <AccountInsertRecord xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects">
      <Currency_x0020_Code>usd</Currency_x0020_Code>
      <Current_x0020_Volume>9</Current_x0020_Volume>
      <Customer_x0020_Account_x0020_Group>Sold-To Party</Customer_x0020_Account_x0020_Group>
    >
      <Location>Location_1</Location>
      <Main_x0020_Phone_x0020_Number>4256543212</Main_x0020_Phone_x0020_Number>
      <Name>Name_Surname</Name>
      <Party_x0020_Name>test_party</Party_x0020_Name>
      <Primary_x0020_Address_x0020_Id>1212 street</Primary_x0020_Address_x0020_Id>
    </AccountInsertRecord>
  </ArrayOfAccountInsertRecord>
</Insert>
```

See [Message Schemas for Business Component Operations](#) for more information about the schema for the request message.

The orchestration consumes the request message and passes it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration. For example, the response from Siebel system for the

above request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<InsertResponse xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/
Account/Operation">
  <InsertResult>
    <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">1-8ANYV</stri
ng>
  </InsertResult>
</InsertResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while performing an operation on business component using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Business Components with MVG Fields Using BizTalk Server

This section provides instructions on performing operation on a business component containing multi-value fields. To demonstrate an end-to-end operation on such business components, you need to perform:

- An INSERT operation on a parent business component
- An INSERT operation on a child business component
- An ASSOCIATE operation on the multi-value links between the parent and child business component
- A Query_[MVG_Child_Business_Comp] operation on a record of parent business component

For more information about how the Siebel adapter supports operations on business components, see [Operations on Business Components](#). For more information about the structure of SOAP messages for performing these operations, see [Message Schemas for Business Component Operations](#).

How to Perform Operations on a Business Component with Multi-value Fields?

Performing an operation on a Siebel system using the Siebel adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform an operation on a business component, these tasks are:

1. Create a BizTalk project and generate schema for all the operations you want to invoke on the business component. As described earlier, to perform operations on a business component with multi-value fields, you must generate schema for Insert operation on the parent and child business components, Associate operation on the multi-value links between the parent and child business components, and a Query operation on the parent business component.
2. Create messages in the BizTalk project for sending and receiving messages from the Siebel system.
3. Create orchestration to invoke the different operations in the Siebel system.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, MVLDemo, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to associate the parent business component, **Account** to a child business component, **Contact**, we will generate the schema for the following:

- INSERT operation on the **Account** business component. See [Performing Operations on Business Components Using BizTalk Server](#).
- INSERT operation on the **Contact** business component.
- ASSOCIATE operation on the **Account** business component.
- QUERY_CONTACT operation on the **Account** business component.

See [Retrieving Metadata for Siebel Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create four sets of request and response messages, one for each operation you will invoke on the Siebel system. Each set must have a request message and a response message. As an example, the following procedure provides instructions to create a request and response message for the Associate operation. You must perform similar steps to create messages for other operations.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the orchestration view the BizTalk project, if not already open. Click **View**, point to **Other Windows**, and click **Orchestration View**.
2. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
3. Right-click the newly create message and select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type AccountAssociate_Request .
Message Type	From the drop-down list, expand Schemas , and select <i>MVLDemo.SiebelBindingSchema.Associate</i> , where <i>MVLDemo</i> is the name of your BizTalk project. <i>SiebelBindingSchema</i> is the schema generated for the invoking the <i>Associate</i> operation on <i>Account</i> business component.

5. Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type AccountAssociate_Response .
Message Type	From the drop-down list, expand Schemas , and select <i>MVLDemo.SiebelBindingSchema.AssociateResponse</i> .

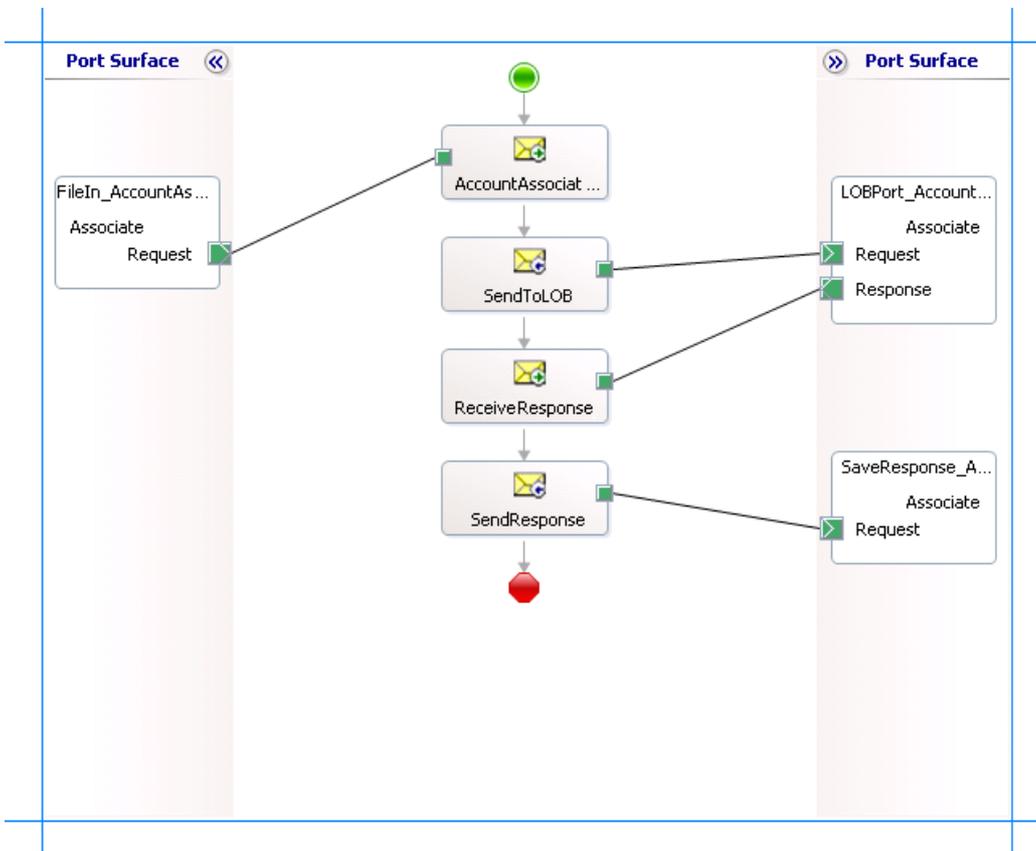
Setting up the Orchestrations

You must now set up orchestrations to use BizTalk Server for performing operations on the Account and Contact business components.

- Set up an orchestration to perform an INSERT operation on the **Account** business component. See [Performing Operations on Business Components Using BizTalk Server](#).
- Set up an orchestration to perform an INSERT operation on the **Contact** business component. This is similar to the INSERT operation on the **Account** business component.
- Set up an orchestration to perform an ASSOCIATE operation on the **Account** business component.
- Set up an orchestration to perform a QUERY_CONTACT operation on the **Account** business component.

This topic will demonstrate how to set up an orchestration for the ASSOCIATE operation on the **Account** business component. You must perform similar tasks to set up the remaining orchestrations as well.

The orchestration for the ASSOCIATE operation on the Account business component looks like:



The following section provides information on how to set up this orchestration by dropping message shapes, ports, linking messages to schemas, etc. You must perform similar tasks to set up the other orchestrations as well.

Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Shape Type	Properties
AccountAssociateXML	Receive	<ul style="list-style-type: none"> Set Name to <i>AccountAssociateXML</i> Set Activate to <i>True</i>
SendToLOB	Send	<ul style="list-style-type: none"> Set Name to <i>SendToLOB</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the *Port* column are the names of the ports as displayed in the orchestration.

Port	Properties
------	------------

FileIn_AccountAssociate	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn_AccountAssociate</i> • Set Type to <i>FileInAccountAssociateType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort_AccountAssociate	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort_AccountAssociate</i> • Set Type to <i>LOBPortAccountAssociateType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse_AccountAssociate	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse_AccountAssociate</i> • Set Type to <i>SaveResponseAccountAssociateType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values to be set to specify messages for action shapes and linking them to the ports. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Properties
AccountAssociateXML	<ul style="list-style-type: none"> • Set Message to <i>AccountAssociate_Request</i> • Set Operation to <i>FileIn_AccountAssociate.Associate.Request</i>
SendToLOB	<ul style="list-style-type: none"> • Set Message to <i>AccountAssociate_Request</i> • Set Operation to <i>LOBPort_AccountAssociate.Associate.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>AccountAssociate_Response</i> • Set Operation to <i>LOBPort_AccountAssociate.Associate.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse_AccountAssociate.Associate.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=102360>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Siebel system. You can have the same port for all the four orchestrations.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Siebel system. You can have the same port for all the four orchestrations.
 - Define physical WCF-Custom or WCF-Siebel send ports to send messages to the Siebel system. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#). You must have different ports for all the four orchestrations.

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Administration Console to create send ports (for outbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for associating a parent business component to the child business component. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The four WCF-Custom or WCF-Siebel send ports, one each for sending messages to the Siebel system, are running
- The four BizTalk orchestrations for the different operations are running

Executing the Operation

You must drop the request messages to the FILE receive port. The schema for the request messages must conform to the schema you generated earlier in the topic. See [Message Schemas for Business Component Operations](#) for more information about the schema for the request message for different operations. You must drop the request messages in the following order:

- Drop a request message to insert a record into the **Account** business component. The request message will be similar to the one dropped for inserting a record in the Account business component in the topic [Performing Operations on Business Components Using BizTalk Server](#). The orchestration consumes the message and sends it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration.
- Drop a request message to insert a record into the **Contact** business component. The request message will be similar to

the one dropped for inserting a record in the Account business component in the topic [Performing Operations on Business Components Using BizTalk Server](#). The orchestration consumes the message and sends it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration.

- Drop a request message to perform an ASSOCIATE operation on the **Account** business component. This will associate the parent and child business components based on the search expression and the name of the multi-valued field you specify in the input XML. Note that:
 - The parent search expression in the ASSOCIATE operation must match a unique record in the parent table.
 - The child search expression in the ASSOCIATE operation must match a unique record in the child table.

For example, the request message to perform an Associate operation on the Account business component is:

```
<Associate xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Operation">
  <ViewMode>3</ViewMode>
  <ParentSearchExpr>[Name] LIKE "SampleName1"</ns0:ParentSearchExpr>
  <ParentMVGField>Bill To First Name</ns0:ParentMVGField>
  <ChildSearchExpr>[First Name] LIKE "SampleName2"</ns0:ChildSearchExpr>
</Associate>
```

The orchestration consumes the message and sends it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration. For example, the response for the above request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<AssociateResponse xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Operation">
  <AssociateResult>
    <ChildID xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects">1-8A009</ChildID>
    <ParentID xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects">1-8ANZ5</ParentID>
  </AssociateResult>
</AssociateResponse>
```

- Drop a request message to perform a QUERY_CONTACT operation on the **Account** business component. For example, the request message for a QUERY_CONTACT operation is:

```
<Query_Contact xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Operation">
  <ViewMode>3</ViewMode>
  <ParentSearchExpr>[Name] LIKE "SampleName1"</ParentSearchExpr>
  <ParentMVGField>Bill To First Name</ParentMVGField>
  <ContactQueryInputRecord>
    <SearchExpr xmlns:ns1="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects">[Id] LIKE '*'</SearchExpr>
  </ContactQueryInputRecord>
</Query_Contact>
```

The orchestration consumes the message and sends it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration.

See [Message Schemas for Business Component Operations](#) for more information about the schema for the request message.

Possible Exceptions

For information about the exceptions you might encounter while performing an operation on business component with multivalued fields using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Business Components with Picklist Fields Using BizTalk Server

A Siebel picklist field type constitutes a collection of possible values from which client may specify a certain value to be passed onto the Siebel system. In other words, a picklist field contains a list of accepted values for a field. For more information about picklist and their types, refer to Siebel documentation. For more information about how the Siebel adapter supports operations on business components with picklist fields, see [Operations on Business Components](#).

When you generate metadata for a business component containing a static bounded picklist field (a type of picklist), the accepted values for the picklist are also published as part of the metadata. If you are inserting a value into a picklist field, you must specify a value that is published in the metadata.

How to Perform Operations on Business Components with Picklist Fields?

Performing an operation on a Siebel system using the Siebel adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform an operation on a business component with picklist field, these tasks are:

1. Create a BizTalk project and generate schema to perform an operation on a business component containing picklist fields.
2. Create messages in the BizTalk project for sending and receiving messages from the Siebel system.
3. Create orchestration to invoke an operation in the Siebel system.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, SiebelPicklist, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to invoke operations on business components with picklist fields, we will generate schema for the **Insert** operation for the **Account** business component. The **Account** business component has a static picklist, *Survey Type*.

See [Retrieving Metadata for Siebel Operations in Visual Studio](#) for more information about how to generate schema.

When you generate the metadata for the Insert operation for the **Account** business component, you get a separate .xsd file containing the picklist fields and their possible values. Note that the .xsd contains only the values for the static picklists, including *Survey Type*.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Siebel system and the other to receive a response.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the orchestration view the BizTalk project, if not already open. Click **View**, point to **Other Windows**, and click **Orchestration View**.

- In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
- Right-click the newly create message and select **Properties Window**.
- In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .
Message Type	From the drop-down list, expand Schemas , and select <i>SiebelPicklist.SiebelBindingSchema.Insert</i> , where <i>SiebelPicklist</i> is the name of your BizTalk project. <i>SiebelBindingSchema</i> is the schema generated for invoking the <i>Insert</i> operation on <i>Account</i> business component.

- Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

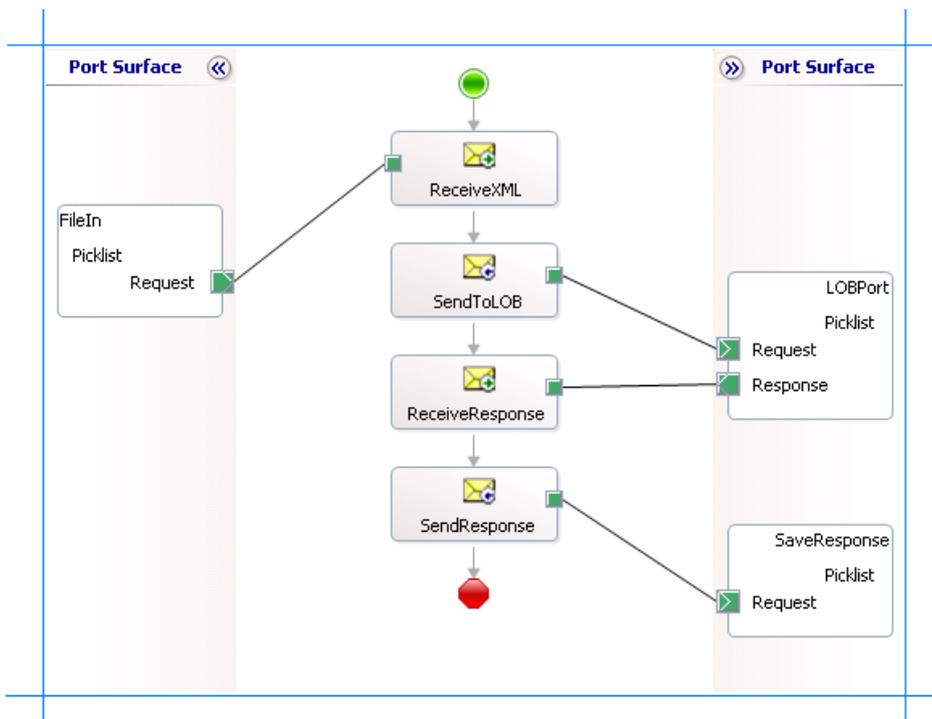
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and select <i>SiebelPicklist.SiebelBindingSchema.InsertResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an Insert operation on a Siebel business component with picklist fields. In this orchestration, you drop a request message at a defined receive location. The Siebel adapter consumes this message and passes it on to the Siebel system. The response from the Siebel system is saved to another location. A typical orchestration for performing operations on Siebel business components would contain:

- Send and receive shapes to send messages to Siebel and receive responses.
- A one-way receive port to receive request messages to send to Siebel.
- A two-way send port to send request messages to Siebel and receive responses.
- A one-way send port to send the responses from Siebel to a folder.

A sample orchestration for the *Insert* operation on an *Account* business component resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Shape Type	Properties
ReceiveXML	Receive	<ul style="list-style-type: none">Set Name to <i>ReceiveXML</i>Set Activate to <i>True</i>
SendToLOB	Send	<ul style="list-style-type: none">Set Name to <i>SendToLOB</i>
ReceiveResponse	Receive	<ul style="list-style-type: none">Set Name to <i>ReceiveResponse</i>Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none">Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the *Port* column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none">Set Identifier to <i>FileIn</i>Set Type to <i>FileInType</i>Set Communication Pattern to <i>One-Way</i>Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none">Set Identifier to <i>LOBPort</i>Set Type to <i>LOBPortType</i>Set Communication Pattern to <i>Request-Response</i>Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none">Set Identifier to <i>SaveResponse</i>Set Type to <i>SaveResponseType</i>Set Communication Pattern to <i>One-Way</i>Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values to be set to specify messages for action shapes and linking them to the ports. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Properties
-------	------------

ReceiveXml	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.Picklist.Request</i>
SendToLOB	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Picklist.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Picklist.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.Picklist.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=102360>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Siebel system.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Siebel system.
 - Define a physical WCF-Custom or WCF-Siebel send port to send messages to the Siebel system. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Administration Console to create send ports (for outbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for performing an *Insert* operation on the *Account* business component in Siebel. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-Siebel send port to send messages to the Siebel system is running
- The BizTalk orchestration for the operation is running

Executing the Operation

You must drop the request message to the FILE receive location. The schema of the request message must conform to the schema you generated earlier in this topic. See [Message Schemas for Business Component Operations](#) for more information about the schema for the request messages.

To insert a value to the picklist field, look at the generated schema to determine the list of acceptable values for the picklist. Make sure the request message has an element to insert value to the picklist field. For example, the request message must contain the following element to insert value to the *Survey Type* picklist.

```
<Survey_x0020_Type>1</Survey_x0020_Type>
```

Here, "1" is an acceptable value for the Survey Type pick list.

An example request message that contains a picklist parameter is:

```
<Insert xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/
Operation">
  <ArrayOfAccountInsertRecord>
    <AccountInsertRecord xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects
">
      <Currency_x0020_Code>usd</Currency_x0020_Code>
      <Current_x0020_Volume>9</Current_x0020_Volume>
      <Customer_x0020_Account_x0020_Group>Sold-To Party</Customer_x0020_Account_x0020_Group
>
      <Location>Location_1</Location>
      <Main_x0020_Phone_x0020_Number>4256543212</Main_x0020_Phone_x0020_Number>
      <Name>Name_Surname</Name>
      <Party_x0020_Name>test_party</Party_x0020_Name>
      <Primary_x0020_Address_x0020_Id>1212 street</Primary_x0020_Address_x0020_Id>
      <Survey_x0020_Type>1</Survey_x0020_Type>
    </AccountInsertRecord>
  </ArrayOfAccountInsertRecord>
</Insert>
```

The orchestration consumes the request message and passes it to the Siebel system. The response from the Siebel system is saved at the other FILE location defined as part of the orchestration.

Note

You may also try an insert an invalid value for the picklist. In that case, you must get a `TargetSystemException`.

Possible Exceptions

For information about the exceptions you might encounter while performing an operation on business component with picklist fields using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Business Service Methods Using BizTalk Server

A Siebel business service is a collection of business methods that can be directly invoked in Siebel. For more information about how the Siebel adapter supports invoking business services in a Siebel system, see [Operations on Business Services](#). For more information about the structure of SOAP messages for performing business service operations, see [Message Schemas for Business Service Operations](#).

How to Invoke Business Services?

Performing an operation on a Siebel system using the Siebel adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a business service, these tasks are:

1. Create a BizTalk project and generate schema for the business service method you want to invoke.
2. Create messages in the BizTalk project for sending and receiving messages from the Siebel system.
3. Create orchestration to invoke a business service method in the Siebel system.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, BusinessService, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to invoke a business service method, we will generate schema for the **Execute** method exposed by the **TimeStamp** business service. See [Retrieving Metadata for Siebel Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must link the schema you generated in the first step to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create two messages—one to send a request to the Siebel system and the other to receive a response.

Perform the following steps to create messages and link them to the schema:

To create messages and link to schema

1. Open the orchestration view the BizTalk project, if not already open. Click **View**, point to **Other Windows**, and click **Orchestration View**.
2. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
3. Right-click the newly create message and select **Properties Window**.
4. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Request .

Message Type	From the drop-down list, expand Schemas , and select <i>BusinessService.SiebelBindingSchema.Execute</i> , where <i>BusinessService</i> is the name of your BizTalk project. <i>SiebelBindingSchema</i> is the schema generated for invoking the <i>Execute</i> business service method.
--------------	--

5. Repeat the previous step to create a new message. In the **Properties** pane for the new message, do the following:

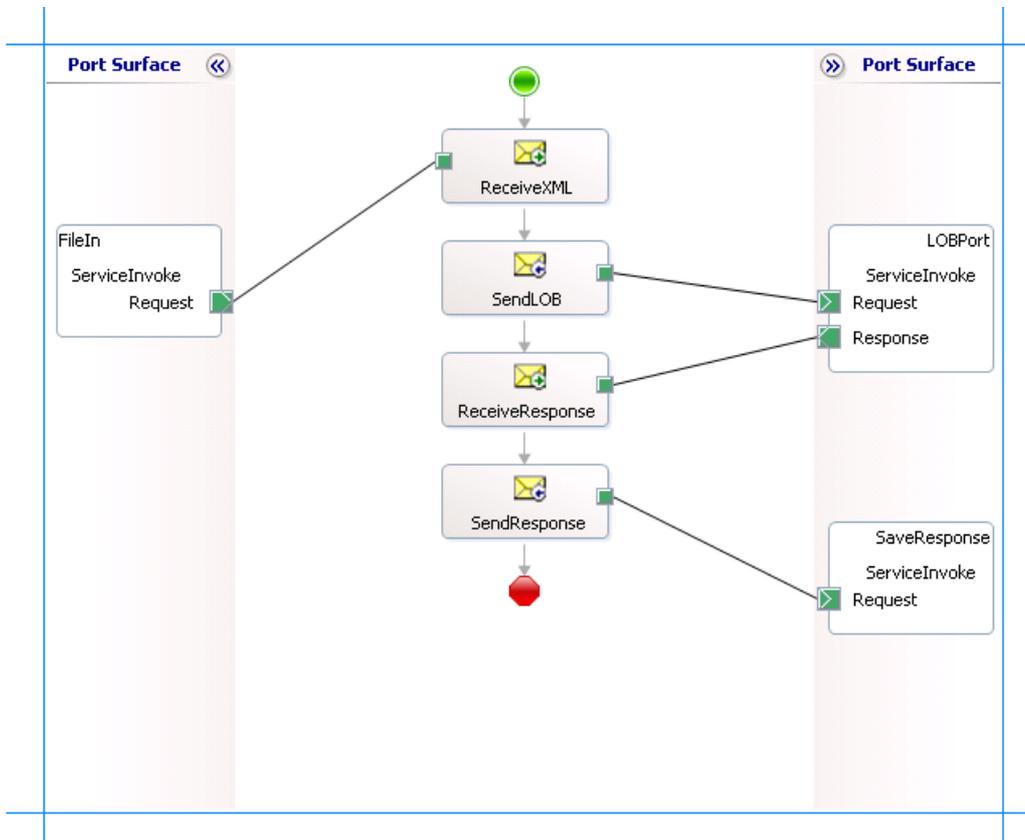
Use this	To do this
Identifier	Type Response .
Message Type	From the drop-down list, expand Schemas , and select <i>BusinessService.SiebelBindingSchema.ExecuteResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for invoking a business service. In this orchestration, you drop a request message at a defined receive location. The Siebel adapter consumes this message and passes it on to the Siebel system. The response from the Siebel system is saved to another location. A typical orchestration for invoking Business Service methods would contain:

- Send and receive shapes to send messages to Siebel and receive responses.
- A one-way receive port to receive request messages to send to Siebel.
- A two-way send port to send request messages to Siebel and receive responses.
- A one-way send port to send the responses from Siebel to a folder.

A sample orchestration for calling the *Execute* method for the *TimeStamp* business service resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveXML	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveXML</i> • Set Activate to <i>True</i>
SendToLOB	Send	<ul style="list-style-type: none"> • Set Name to <i>SendToLOB</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the *Port* column are the names of the ports as displayed in the orchestration.

Port	Properties
FileIn	<ul style="list-style-type: none"> • Set Identifier to <i>FileIn</i> • Set Type to <i>FileInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
SaveResponse	<ul style="list-style-type: none"> • Set Identifier to <i>SaveResponse</i> • Set Type to <i>SaveResponseType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values to be set to specify messages for action shapes and linking them to the ports. The names listed in the *Shape* column are the names of the message shapes as displayed in the orchestration above.

Shape	Properties
ReceiveXML	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>FileIn.ServiceInvoke.Request</i>

SendToLOB	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.ServiceInvoke.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.ServiceInvoke.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>SaveResponse.ServiceInvoke.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=102360>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the Siebel system.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the Siebel system.
 - Define a physical WCF-Custom or WCF-Siebel send port to send messages to the Siebel system. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the Siebel Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file containing information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Administration Console to create send ports (for outbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for executing the *Execute* method on the *TimeStamp* business service in Siebel. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.

- The WCF-Custom or WCF-Siebel send port to send messages to the Siebel system is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

You must drop the request message to the FILE receive location. The schema of the request message must confirm to the schema you generated earlier in this topic. See [Message Schemas for Business Service Operations](#) for more information about the schema for invoking a business service. For example, the request message to invoke the *Execute* method on the *TimeStamp* business service is:

```
<Execute xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/TimeStamp/Operation" />
```

The orchestration consumes the request message and passes it to the Siebel system. The response from the Siebel system is saved at the FILE send location. The response for the above request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<ExecuteResponse xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/TimeStamp/Operation">
  <ExecuteResult>
    <Time xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/TimeStamp">2007-11-25T20:42:11.0000000</Time>
  </ExecuteResult>
</ExecuteResponse>
```

Possible Exceptions

For information about the exceptions you might encounter while performing an operation on business service using BizTalk Server, see [Exceptions and Error Handling](#).

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the bindings file. Once you generate a bindings file, you can import the configuration settings from the file so that you do not need to create the send ports, receive ports, etc. for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Business Service Methods with Integration Objects

The Siebel adapter enables adapter clients to invoke business service methods that work with integration objects. These business services typically have IN, OUT, or IN OUT parameters of "hierarchy" data type to send or receive integration object data.

The Siebel adapter exposes these hierarchical types as strings. These string values are essentially an XML string encapsulated in an XML CDATA section. The XML string is compatible with the XML schema of the integration object the adapter client is trying to send or receive.

Sample Based On This Topic

A sample, `SiebelAdapterIntegrationObjects`, based on this topic is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Creating an Orchestration to Invoke Business Service Methods with Integration Objects

Creating an orchestration to invoke a business service method that takes integration object parameters is similar to the orchestration to invoke any other business service, as described in [Invoking Business Service Methods Using BizTalk Server](#).

The difference lies in the request message that you drop for the orchestration. This difference is because of the following:

- The schema for the request message is different because you invoke a different business service.
- The request message contains the XML string for the integration object. This XML is encapsulated in a CDATA section. You must first generate the schema for the integration object and then create an XML that conforms to the schema. This XML must be passed within the CDATA section of the request message sent to the adapter.

After you have generated the XML that conforms to the integration object schema and included it in the request message, you must drop the request message at a FILE location, just like you do for any other orchestration. Look for the response message in the other FILE location.

Request and Response Messages for Invoking Business Service with Integration Objects

As mentioned before, the only difference for invoking a business service that takes integration object parameters is the request message sent to the adapter. This section provides instructions on the steps you must perform to create the request message.

For better understanding, take a Siebel business service 'EAI Siebel Adapter' as an example. The 'EAI Siebel Adapter' business service operates on a Siebel integration object, 'Sample Account'. You must perform the following tasks to create the request message to invoke a method exposed by the 'EAI Siebel Adapter' business service:

- **Generate the schema for the EAI Siebel Adapter business service.** You must use the Consume Adapter Service Add-in to generate the schema. Once you generate the schema, generate the XML that conforms to the schema.
- **Generate the schema for the integration object.** Use Siebel Tools to generate schema for an integration object. From the Siebel Tools interface, select the integration object, for example, Sample Account, and click **Generate Schema**. While generating the schema, make sure:
 - The **Select a Business Service from the list** drop-down has the value "EAI XML XSD Generator".
 - The **Select an envelope type from the list** drop-down has the value "Siebel Message Envelope".

For more information, see the Siebel documentation.

- **Create an XML message that conforms to schema of the integration object.** A sample XML message generated for the 'Sample Account' integration object looks like:

```
<?xml version="1.0" encoding="UTF-16"?>
<SiebelMessage MessageId="" IntObjectName="Sample Account" MessageType="Integration Object" IntObjectFormat="Siebel Hierarchical">
  <ListOfSampleAccount>
```

```

<Account>
  <CurrencyCode>USD</CurrencyCode>
  <Location>Redmond</Location>
  <Name>IntegrationObjectTest</Name>
</Account>
</ListOfSampleAccount>
</SiebelMessage>

```

- **Pass this XML message as the value for the CDATA element in the request message for the business service method.** A sample request message that contains the preceding XML message within a CDATA element looks like the following. This sample request is to invoke the Insert method for the 'EAI Siebel Adapter' business service.

```

<Insert xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/EAI_x0020_
Siebel_x0020_Adapter/Operation">
  <InsertRequestRecord />
  <InsertInOutRecord>
    <SiebelMessage xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices
/EAI_x0020_Siebel_x0020_Adapter">
      <![CDATA[ <?xml version="1.0" encoding="UTF-16"?>
        <SiebelMessage MessageId="" IntObjectName="Sample Account" MessageType="Integ
ration Object" IntObjectFormat="Siebel Hierarchical">
          <ListOfSampleAccount>
            <Account>
              <CurrencyCode>USD</CurrencyCode>
              <Location>Redmond</Location>
              <Name>IntegrationObjectTest</Name>
            </Account>
          </ListOfSampleAccount>
        </SiebelMessage>
      ]]>
    </SiebelMessage>
  </InsertInOutRecord>
</Insert>

```

The response from Siebel for the above request message resembles the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<InsertResponse xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/EAI
_x0020_Siebel_x0020_Adapter/Operation">
  <InsertResult>
    <ErrorCode xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/EAI
_x0020_Siebel_x0020_Adapter">0x0</ErrorCode>
    <ErrorContextIntComp xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessSe
rvices/EAI_x0020_Siebel_x0020_Adapter" />
    <ErrorContextSearchSpec xmlns="http://Microsoft.LobServices.Siebel/2007/03/Busines
sServices/EAI_x0020_Siebel_x0020_Adapter" />
    <ErrorSymbol xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/E
AI_x0020_Siebel_x0020_Adapter" />
    <OMErrorCode xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/E
AI_x0020_Siebel_x0020_Adapter" />
    <OMErrorSymbol xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices
/EAI_x0020_Siebel_x0020_Adapter" />
    <PrimaryRowId xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/
EAI_x0020_Siebel_x0020_Adapter">1-6SPSJ</PrimaryRowId>
  </InsertResult>
</InsertInOutRecord>
  <SiebelMessage xmlns="http://Microsoft.LobServices.Siebel/2007/03/BusinessServices

```

```
/EAI_x0020_Siebel_x0020_Adapter">
  <![CDATA[ <?xml version="1.0" encoding="UTF-16"?>
    <SiebelMessage MessageId="" IntObjectName="Sample Account" MessageType="Integ
ration Object" IntObjectFormat="Siebel Hierarchical">
      <ListOfSampleAccount>
        <Account>
          <CurrencyCode>USD</CurrencyCode>
          <Location>Redmond</Location>
          <Name>IntegrationObjectTest</Name>
        </Account>
      </ListOfSampleAccount>
    </SiebelMessage>
  ]]>
</InsertInOutRecord>
</InsertResponse>
```

Using BizTalk features, adapter clients can also perform additional validation of the integration object IN and OUT parameter against the integration object schema (obtained from Siebel Tools.)

See Also

Concepts

[Development](#)

Other Resources

[Developing BizTalk Applications](#)

Developing Applications by Using the WCF Service Model

WCF provides a programming model called the WCF service model, which, in part, helps address some of the limitations of another programming model—the WCF channel model.

At the lowest level, the Windows Communication Foundation (WCF) presents the WCF channel model in which clients invoke operations on a service by exchanging SOAP messages over a channel established between client and service endpoints. The WCF channel model exposes data types and methods that enable you to operate directly on the WCF channel architecture. The WCF channel model provides you with direct control over the contents of the SOAP messages you create and over the way both your application and the Siebel adapter consume them. However, creating well-formed SOAP messages to send over a channel and validating the reply messages returned can be a detailed and exacting task.

The WCF service model, however, involves the use of proxy classes to invoke operations on a target service or to receive operations from a client. The Siebel adapter exposes the Siebel system as a WCF service on which you can invoke operations.

- The proxy class used to invoke operations on a target service is called a WCF client class. This class models the operations exposed by a service as .NET methods with strongly-typed parameters. By using the WCF service model, you can invoke the operations exposed by the Siebel adapter as .NET methods on the WCF client. For more information about WCF clients, see "WCF Client Overview" at <http://go.microsoft.com/fwlink/?LinkId=91458>.

You use tools to generate a WCF client class and associated helper code from the service metadata that the Siebel adapter exposes. You can use either of the following tools:

- The ServiceModel Metadata Utility Tool (svcutil.exe), which ships with WCF
- The Add Adapter Service Reference Visual Studio Plug-in, which ships with the Siebel adapter

The Add Adapter Service Reference Plug-in is integrated with the Visual Studio design experience and presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations exposed by the adapter. For more information about how to generate a WCF client, see [Generating a WCF Client for Siebel Artifacts](#)

Why Choose the WCF Service Model or the WCF Channel Model?

Because it presents a model that is familiar to .NET programmers and hides the underlying complexities of SOAP message exchange over a channel, the WCF service model is often the best choice to develop programming solutions for the Siebel adapter. However, there are scenarios in which the WCF channel model might be a better choice. For example, serializing and de-serializing between the XML representation of objects in a SOAP message and the .NET types used to represent them in the WCF service model involves reading the entire message into memory.

The WCF channel model provides support for XML node-level streaming on all operations. In node-level streaming, only each node of the XML message is held in memory at any one time. For certain operations, for example, if you are performing queries that return large result sets, the WCF channel model might be a better choice for your application. For more information about using the WCF channel model, see [Developing Applications by Using the WCF Channel Model](#).

The topics in this section contain information, procedures, and examples to help you create and use the WCF service model to develop applications by using the Siebel adapter.

In This Section

- [Overview of Using the WCF Service Model with the Adapter](#)
- [Metadata and the WCF Service Model](#)
- [Generating a WCF Client for Siebel Artifacts](#)
- [Configuring a WCF Client for a Siebel System](#)
- [Performing Operations on Business Components by Using the WCF Service Model](#)
- [Performing Operations on Business Components with MVG Fields by Using the WCF Service Model](#)

- [Invoking Business Service Methods by Using the WCF Service Model](#)

See Also

Concepts

[Development](#)

Overview of Using the WCF Service Model with the Adapter

The Microsoft BizTalk Adapter for Siebel eBusiness Applications exposes a Siebel system as a WCF service. To perform operations on Siebel system artifacts, for example to invoke a method of a Siebel business service, you invoke an operation on the adapter, which, in turn, performs the operation on the Siebel system. Your code therefore acts as a client to the WCF service presented by the adapter.

In the Windows Communication Foundation (WCF) service model, the service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The Siebel adapter and WCF provide tools that enable you to generate this interface for targeted operations from the metadata that the adapter exposes. These tools also create a WCF client class that can be used to invoke the operations exposed in the service interface. A client application can call the methods of the WCF client class to invoke operations on the adapter.

The following section explains how to use the WCF service model to invoke operations with a WCF client.

Invoking Operations on the Siebel System with a WCF Client

To use the WCF service model to invoke operations on the Siebel adapter, you must first generate a WCF client class for the target operations. You can then create an instance of this class, a WCF client, and call its methods to perform these operations on the Siebel system.

To invoke operations on the Siebel adapter

1. Generate a WCF client class and helper code. Use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class targeted at the Siebel system artifacts with which you want to work. For more information about how to generate a WCF client, see [Generating a WCF Client for Siebel Artifacts](#).
2. Create a WCF client instance and configure the WCF client. Configuring the WCF client involves specifying the binding and endpoint address (connection URI) that the client will use. You can do this either imperatively in code or declaratively in configuration. For more information about how to configure the WCF client, see [Configuring a WCF Client for a Siebel System](#). The following code creates a WCF client that targets the Siebel TimeStamp business service. It also sets the credentials for the Siebel system. The WCF client is initialized from configuration.

```
BusinessServices_TimeStamp_OperationClient client =
    new BusinessServices_TimeStamp_OperationClient("SiebelBinding_BusinessServices_Tim
eStamp_Operation");

client.ClientCredentials.UserName.UserName = "YourUserName";
client.ClientCredentials.UserName.Password = "YourPassword";
```

3. Open the WCF client.

```
client.Open();
```

4. Invoke methods on the WCF client created in step 2 to perform operations on the Siebel system. The following code invokes the **Execute** method of the WCF client to invoke the **Execute** method of the TimeStamp business service on the Siebel system.

```
// Create a parameter to hold the results and then invoke the Execute method of the Ti
meStamp business service.
microsoft.lobservices.siebel._2007._03.BusinessServices.TimeStamp.ExecuteResponseReco
rder;
er = client.Execute();
```

5. Close the WCF client.

```
client.Close();
```

For more information about invoking Siebel business service methods, see [Invoking Business Service Methods by Using the WCF Service Model](#)

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Metadata and the WCF Service Model

In the WCF service model, you use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutile.exe) to generate a proxy class—the WCF client class—through which your code can invoke operations on the Microsoft BizTalk Adapter for Siebel eBusiness Applications.

These tools use the metadata exposed by the adapter to generate:

- An annotated .NET interface that represents the service contract for target operations. This interface is called the WCF service contract.
- Annotated classes that represent the supporting message contracts, operation contracts, and data contracts for the service contract.
- A WCF client class whose methods can be used to invoke the service methods (operations) exposed by the WCF service contract.

For help in understanding the structure of this generated code, see "Understanding Generated Client Code" at <http://go.microsoft.com/fwlink/?LinkId=98365>. This topic specifically describes code that svcutil.exe generates, but its content is also applicable to the code that the Add Adapter Service Reference Plug-in generates.

For information about how to generate a WCF client class for target operations and about the differences between svcutil.exe and the Add Adapter Service Reference Plug-in, see [Generating a WCF Client for Siebel Artifacts](#)

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Generating a WCF Client for Siebel Artifacts

You can use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class targeted at selected operations on Siebel artifacts. You can also use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate the WCF client class; however, the Add Adapter Service Reference Plug-in exposes the functionality of the ServiceModel Metadata Utility Tool through a standard Microsoft Windows interface. It also provides browse and search capabilities that are not available with the svcutil.exe tool, and generates a configuration file based on the binding properties that you select when you connect to the Siebel system.

Generating a WCF Client Class by Using the Add Adapter Service Reference Plug-in

Perform the following steps to generate a WCF client class by using the Add Adapter Service Reference Plug-in.

To generate a WCF client class

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for Siebel Operations in Visual Studio](#) to connect to the Siebel system and browse and search for operations. To create a WCF client class for the operations that you select, be sure that **Client (Outbound operations)** is selected from the **Select contract type** drop-down list (this is the default).
3. After you select all of the operations that you want to target, click **OK** to generate the WCF client class.

The Add Adapter Service Reference Plug-in adds two files to your project:

- The WCF client code file. This file contains the generated WCF client class and helper code for the operations that you selected. The first time you run the Add Adapter Service Reference Visual Studio Plug-in it will generate this file with the default name **SiebelBindingClient.cs**. If you run it again, the next file it generates will be called **SiebelBindingClient1.cs**. The number suffix will increase by 1 for every new file you generate. You can also change the default prefix **SiebelBinding** by entering a different prefix in the **Filename prefix** field of the Add Adapter Service Reference Visual Studio Plug-in before selecting **OK** to generate the file.
- **App.config**. This file contains a binding configuration and client endpoint configurations that are based on the selections you made when you configured the connection for the Add Adapter Service Reference Plug-in. For more information about the contents of the app.config file, see [Configuring a WCF Client for a Siebel System](#).

◆ Important

While using the Add Adapter Service Reference Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Generating a WCF Client Class by Using svcutil.exe

You can use svcutil.exe to generate a WCF client class for your application. You must configure svcutil.exe to use it with the Microsoft BizTalk Adapter for Siebel eBusiness Applications. For more information about configuring and using svcutil.exe with the Siebel adapter, see [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Siebel eBusiness Applications](#).

Svcutil.exe generates the WCF client class in an output file with a default file name of output.cs. You must manually include this file in your Visual Studio project.

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Configuring a WCF Client for a Siebel System

After you have generated the WCF client class, you can create a WCF client (instance) and invoke its methods to consume the Siebel adapter. For information about how to generate the WCF client class and helper code for operations that the Microsoft BizTalk Adapter for Siebel eBusiness Applications exposes, see [Generating a WCF Client for Siebel Artifacts](#).

To create the WCF client, you must specify an endpoint address and a binding. The endpoint address must contain a valid Siebel connection URI, and the binding must be an instance of a Siebel Binding (**SiebelBinding**). For more information about the Siebel connection URI, see [Connecting to the Siebel System in Visual Studio](#).

You can specify the Siebel Binding and the endpoint address in your code or in a configuration file. When you use the Add Adapter Service Reference Visual Studio Plug-in to generate the WCF client class, a configuration file (app.config) is also created for your project. This file contains configuration settings that reflect the binding properties and connection information (except credentials) that you specified when you connected to the Siebel system with the Add Adapter Service Reference Plug-in.

Specifying the Binding and Endpoint Address in Code

The following code shows how to create a WCF client by specifying the binding and endpoint address in code. It is good practice to specify the Siebel credentials by using the **ClientCredentials** property of the WCF client rather than in the connection URI supplied for the endpoint address.

```
// A WCF client that targets the TimeStamp business service is created
// by using a binding object and endpoint address
SiebelBinding sblBinding = new SiebelBinding();
EndpointAddress sblAddress = new EndpointAddress("siebel://Siebel_server:1234?SiebelObjectM
anager=obj_mgr&SiebelEnterpriseServer=ent_server&Language=enu");

BusinessServices_TimeStamp_OperationClient client =
    new BusinessServices_TimeStamp_OperationClient(sblBinding, sblAddress);

client.ClientCredentials.UserName.UserName = "SADMIN";
client.ClientCredentials.UserName.Password = "SADMIN";

client.Open();
```

Specifying the Binding and Endpoint Address in a Configuration File

The following code shows how to create a WCF client by specifying the binding and endpoint address in an app.config file.

```
// A WCF client that targets the TimeStamp business service is created
// by specifying the client endpoint information in app.config
BusinessServices_TimeStamp_OperationClient client =
    new BusinessServices_TimeStamp_OperationClient("SiebelBinding_BusinessServices_TimeStam
p_Operation");

client.ClientCredentials.UserName.UserName = "SADMIN";
client.ClientCredentials.UserName.Password = "SADMIN";

client.Open();
```

The App.config File

The following XML shows the configuration file created for the TimeStamp business service by the Add Adapter Service Reference Plug-in. This file contains the client endpoint configuration referenced in the preceding example.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <siebelBinding>
        <binding name="SiebelBinding" closeTimeout="00:01:00" openTimeout="00:01:00"
receiveTimeout="00:10:00" sendTimeout="00:01:00" enableBizTalkCompatibi
```

```

lityMode="false"
        enablePerformanceCounters="false" enableConnectionPooling="true"
        maxConnectionsPerSystem="5" idleConnectionTimeout="00:01:00"
        acceptCredentialsInUri="false" />
    </siebelBinding>
</bindings>
<client>
    <endpoint address="siebel://Siebel_server:1234?SiebelObjectManager=obj_mgr&
SiebelEnterpriseServer=ent_server&Language=enu"
        binding="siebelBinding" bindingConfiguration="SiebelBinding"
        contract="BusinessServices_TimeStamp_Operation" name="SiebelBinding_Busines
sServices_TimeStamp_Operation" />
    </client>
</system.serviceModel>
</configuration>

```

If a project has more than one WCF client, there will be multiple client endpoint entries defined in the configuration file. Each WCF client entry will have a unique name based on its binding configuration and target Siebel artifact; for example, "SiebelBinding_BusinessServices_TimeStamp_Operation ". If you connect multiple times to create the WCF clients in your project, multiple binding configuration entries will be created, one for each connection. These binding configuration entries will be named in the following manner: SiebelBinding, SiebelBinding1, SiebelBinding2, and so on. Each client endpoint entry created during a specific connection will reference the binding entry created during that connection.

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Performing Operations on Business Components by Using the WCF Service Model

You can create a WCF client that targets a Siebel business component. You can then use the WCF client to perform Insert, Update, Query, Delete, Associate, Dissociate, and child record query operations on the business component on the Siebel system. Siebel business objects are surfaced under the Business Objects node in the Add Adapter Service Reference Visual Studio Plug-in. The business components that compose each business object are surfaced under the node corresponding to that object. You can follow the steps in [Overview of Using the WCF Service Model with the Adapter](#) to generate a WCF client for a business component that targets specific operations and use the client to invoke these operations on the business component.

Note

This topic provides information about performing basic operations (Insert, Update, Query, Delete) on business components. For more information about performing Associate, Dissociate, and child record query operations, see [Performing Operations on Business Components with MVG Fields by Using the WCF Service Model](#)

The following code uses a WCF client to perform Insert, Update, and Delete operations on a record in the Account business component of the Account business object. Between each operation, a Query operation is performed to verify the results. The WCF client in this example is initialized from a configuration file that is generated by the Add Adapter Service Reference Plug-in. For an example of a generated configuration file, see [Configuring a WCF Client for a Siebel System](#).

```
using System;
using System.Collections.Generic;
using System.Text;

using System.ServiceModel;
using microsoft.lobservices.siebel._2007._03.BusinessObjects;
using microsoft.lobservices.siebel._2007._03;

namespace Business_Component_Operations
{
    class Program
    {
        static void Main(string[] args)
        {
            BusinessObjects_Account_Account_OperationClient client = null;

            try
            {
                client = new BusinessObjects_Account_Account_OperationClient("SiebelBinding
_BusinessObjects_Account_Account_Operation");
                client.ClientCredentials.UserName.UserName = "YourUserName";
                client.ClientCredentials.UserName.Password = "YourPassword";

                Console.WriteLine("Opening connection to " + client.Endpoint.Address.Uri.Host);

                client.Open();
                AccountInsertRecord[] air = new AccountInsertRecord[1];
                air[0] = new AccountInsertRecord();
                Random r = new Random();
                int count = r.Next();

                air[0].Name = "Anil" + count.ToString();
                Console.WriteLine("Inserting " + air[0].Name + " ...");
                client.Insert(air);

                //Query for the record.
                Console.WriteLine("Querying to check inserted record ...");
                AccountQueryInputRecord aqir = new AccountQueryInputRecord();
```

```

short viewmode = new short(); viewmode = 3;
string[] fields = new string[2];
fields[0] = "Name";
fields[1] = "Id";
aqir.QueryFields = fields;
aqir.SearchExpr = "[Name] LIKE \"Anil*\"";
AccountQueryRecord[] aqr = client.Query(viewmode, aqir);
Console.WriteLine(aqr.Length + " records returned");
string[] ids = new string[1];
AccountUpdateRecord[] aur = new AccountUpdateRecord[1];
aur[0] = new AccountUpdateRecord();
foreach (AccountQueryRecord rec in aqr)
{
    Console.WriteLine("Name: " + rec.Name);
    Console.WriteLine("Id: " + rec.Id);
    Console.WriteLine("");
    ids[0] = rec.Id;
    aur[0].Name = rec.Name;
    aur[0].Id = rec.Id;
}

//Update it.

aur[0].Name = "Anil" + count.ToString() + "modified";
Console.WriteLine("Update the record to " + aur[0].Name + " ...");
client.Update(viewmode, aur);
//query again
aqr = client.Query(viewmode, aqir);
Console.WriteLine(aqr.Length + " records returned");
foreach (AccountQueryRecord rec in aqr)
{
    Console.WriteLine("Name: " + rec.Name);
    Console.WriteLine("Id: " + rec.Id);
    Console.WriteLine("");
}

//Delete it.
Console.WriteLine("Deleting record ...");
client.Delete(viewmode, ids, "");

//Check again.
Console.WriteLine("Check that " + aur[0].Name + " is not present");
aqr = client.Query(viewmode, aqir);

Console.WriteLine(aqr.Length + " records returned");
foreach (AccountQueryRecord rec in aqr)
{
    Console.WriteLine(rec.Name);
    Console.WriteLine(rec.Id);
}

// Wait for <RETURN> to end sample.
Console.WriteLine("\nHit <RETURN> to finish");
Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
finally
{
    // Close the client.
    if (client != null)
    {
        if (client.State == CommunicationState.Opened)

```

```
        client.Close();
    else
        client.Abort();
    }
}
}
```

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Performing Operations on Business Components with MVG Fields by Using the WCF Service Model

The Microsoft BizTalk Adapter for Siebel eBusiness Applications surfaces operations to associate, dissociate, and query child records for business components that have multivalued group (MVG) fields. A sample that demonstrates how to work with MVG fields in the WCF service model is included in the samples provided with the Siebel adapter. For more information, see [Samples](#).

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Invoking Business Service Methods by Using the WCF Service Model

You can create a WCF client that targets methods of Siebel business services. You can then use the WCF client to invoke these methods on the Siebel system. Siebel business services are surfaced under the Business Services node in the Add Adapter Service Reference Visual Studio Plug-in. The methods exposed by each business service are surfaced under the node corresponding to that service. You can follow the steps in [Overview of Using the WCF Service Model with the Adapter](#) to generate a WCF client for a business service and to use it to invoke the service's methods.

The following code uses a WCF client to invoke the **Execute** method of the TimeStamp business service. The time stamp, which is returned in the local time of the Siebel server, is then written to the console. The WCF client in this example is initialized from a configuration file generated by the Add Adapter Service Reference Plug-in. For an example of a generated configuration file, see [Configuring a WCF Client for a Siebel System](#).

```
using System;
using System.Collections.Generic;
using System.Text;

using System.ServiceModel;
using microsoft.lobservices.siebel._2007._03.BusinessServices.TimeStamp;

namespace Business_Service
{
    class Program
    {
        static void Main(string[] args)
        {
            BusinessServices_TimeStamp_OperationClient client = null;

            try
            {
                client =
                    new BusinessServices_TimeStamp_OperationClient("SiebelBinding_Business
Services_TimeStamp_Operation");

                client.ClientCredentials.UserName.UserName = "YourUserName";
                client.ClientCredentials.UserName.Password = "YourPassword";
                client.Open();

                ExecuteResponseRecord er = client.Execute();
                Console.WriteLine(er.Time);

                Console.WriteLine("\nHit <RETURN> to finish");
                Console.ReadLine();
            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
            finally
            {
                // Close the client.
                if (client != null)
                {
                    if (client.State == CommunicationState.Opened)
                        client.Close();
                    else
                        client.Abort();
                }
            }
        }
    }
}
```

}

See Also

Concepts

[Developing Applications by Using the WCF Service Model](#)

Developing Applications by Using the WCF Channel Model

This section provides information about creating a channel and the operations that you can perform on Siebel using the channel.

In This Section

- [Creating a Channel](#)
- [Performing Operations on Business Components by Using the WCF Channel Model](#)

Creating a Channel

This section demonstrates how to create a channel for direct messaging with Siebel by providing and consuming XML messages.

```
//create a channel factory, capable of sending a request to Siebel and receiving a reply (I
RequestChannel)
IChannelFactory<IRequestChannel> factory = binding.BuildChannelFactory<IRequestChannel>(new
BindingParameterCollection());

//open factory
factory.Open();

//obtain a channel from the factory by specifying the address you want to connect to
IRequestChannel irc = factory.CreateChannel(address);

//open the channel
irc.Open();

//perform operations
.....
.....
.....

//close the channel
irc.Close();

//close the factory
factory.Close();
```

Once you have created the channel, you can use that channel to perform operations on Siebel.

See Also

Concepts

[Developing Applications by Using the WCF Channel Model](#)

[Performing Operations on Business Components by Using the WCF Channel Model](#)

Performing Operations on Business Components by Using the WCF Channel Model

This section demonstrates how to perform operations on Siebel using the channel created in [Creating a Channel](#).

```
// create binding
SiebelBinding binding = new SiebelBinding();

//set up an endpoint address
EndpointAddress address = new EndpointAddress("siebel://Username=myuser;Password=mypass@mysiebelserver:1234?SiebelObjectManager=SSEObjMgr&SiebelEnterpriseServer=ent771&Language=enu");

//create request channel factory
IChannelFactory<IRequestChannel> factory = binding.BuildChannelFactory<IRequestChannel>(new BindingParameterCollection());

//open factory
factory.Open();

//create request channel using endpoint
IRequestChannel channel = factory.CreateChannel(address);

//open the channel
channel.Open();

// send request message and receive reply
System.Xml.XmlReader readerIn = System.Xml.XmlReader.Create(inputXml);
System.ServiceModel.Channels.Message messageIn = System.ServiceModel.Channels.Message.CreateMessage(MessageVersion.Default, action, readerIn);
System.ServiceModel.Channels.Message messageOut = channel.Request(messageIn);

// get response XML from SOAP message
System.Xml.XmlReader readerOut = messageOut.GetReaderAtBodyContents();

// save output file
XmlDocument doc = new XmlDocument();
doc.Load(readerOut);
doc.Save(outputXml);
Console.WriteLine("XML written out to {0}", outputXml);

// close the channel and the factory
channel.Close();
factory.Close();
```

See Also

Concepts

[Developing Applications by Using the WCF Channel Model](#)

[Performing Operations on Business Components by Using the WCF Service Model](#)

Using the .NET Framework Data Provider for Siebel eBusiness Applications

This section provides instructions on using the .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel). This section provides information about:

- The connection string to connect to a Siebel system using an ADO.NET client.
- The syntax for SELECT and EXEC statements.
- Using the Data Provider for Siebel with SSIS.
- The ADO.NET interfaces that the Data Provider for Siebel extends.

In This Section

- [Extending ADO.NET Interfaces](#)
- [The Data Provider for Siebel Connection String](#)
- [Syntax for a SELECT Statement](#)
- [Syntax for an EXEC Statement](#)
- [Performing a SELECT Query on Business Components](#)
- [Performing an EXECUTE Operation on Business Services](#)
- [Using the Data Provider for Siebel with SSIS](#)

Extending ADO.NET Interfaces

The .NET Framework Data Provider for Siebel eBusiness Applications extends various ADO.NET interfaces. The ADO.NET clients can use these to interface with the Siebel system. This section provides more information about these interfaces.

In This Section

- [SiebelClientFactory](#)
- [SiebelConnection](#)
- [SiebelCommand](#)
- [SiebelParameter](#)
- [SiebelConnectionStringBuilder](#)
- [DbDataReader](#)

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

SiebelClientFactory

An ADO.NET client accesses the Data Provider for Siebel using generic ADO.NET classes and interfaces. To enable this feature, the Data Provider for Siebel inherits the **System.Data.Common.DbProviderFactory** class. The client program consumes the client as follows:

```
DbProviderFactory factory = DbProviderFactories.GetFactory("Microsoft.Data.SiebelClient");  
DbConnection connection = factory.CreateConnection();
```

An alternative approach is as follows:

```
SiebelClientFactory factory = SiebelClientFactory.Instance;  
DbConnection connection = factory.CreateConnection();
```

SiebelClientFactory exists in the namespace Microsoft.Data.SiebelClient.

Supported Members

SiebelClientFactory extends **System.Data.Common.DbProviderFactory**. The following table provides a description of the members that **SiebelClientFactory** overrides.

Name	Description
Instance	This is a member variable. It provides a singleton instance of SiebelClientFactory.
CreateCommand()	Creates an instance of SiebelCommand.
CreateConnection()	Creates an instance of SiebelConnection.
CreateConnectionStringBuilder()	Creates an instance of SiebelConnectionStringBuilder.
CreateParameter()	Creates an instance of SiebelParameter.

See Also

Other Resources

[Extending ADO.NET Interfaces](#)

SiebelConnection

The Data Provider for Siebel accesses the underlying Siebel adapter `Binding`, the `ConnectionFactory`, and `Channel` to connect to the Siebel system. The Data Provider for Siebel implements the `DbConnection` class to support the preceding features.

Using an instance of `Microsoft.Data.SiebelClient.SiebelClientFactory`, a client program can obtain an instance of the `System.Data.Common.DbConnection` class to connect to the Siebel system.

```
//In this example, factory is an instance of SiebelClientFactory
DbConnection connection = factory.CreateConnection();
```

Alternatively, the following approach can be used to create a connection:

```
SiebelConnection connection = new SiebelConnection();
connection.ConnectionString = connectionString;
```

The `SiebelConnection` class inherits from `DbConnection`. It exists in the namespace `Microsoft.Data.SiebelClient`.

Supported Properties

The `SiebelConnection` class supports the following `DbConnection` properties.

Name	Get/Set	Description
ConnectionString	Get and Set	Gets or sets the string used to open the connection.
Database	Get	Gets the name of the current database after a connection is opened, or the database name specified in the connection string before the connection is opened. This should be the Siebel repository name.
DataSource	Get	Gets the name of the Siebel gateway for this connection.
ServerVersion	Get	In the current version of Data Provider for Siebel, this property returns a hard-coded value, which does not present the actual version of the Siebel server.
State	Get	Gets a string that describes the state of the connection. This can contain three possible values: OPEN, BROKEN, or CLOSED.

Supported Methods

The `SiebelConnection` class supports the following `DbConnection` methods.

Name	Description
CreateDbCommand	This protected method provides a new <code>DbCommand</code> instance.
ChangeDatabase	This public method is not supported, and will throw an exception if called.
Open	Opens a connection with the Siebel system by creating a WCF channel.
Close	Closes a connection with the Siebel system by closing a WCF channel.
CreateCommand	Creates a command object.

See Also

Other Resources

SiebelCommand

After establishing a connection with the Siebel system, the Data Provider for Siebel parses the Siebel command strings and command parameters provided by the ADO.NET client and maps the command into a WCF request message. The Data Provider for Siebel then sends the request to the Siebel adapter and obtains the response XML and the body contents from the adapter. The Data Provider for Siebel then uses the `XMLDataReader` to retrieve the relational data from the XML body.

Using an instance of `Microsoft.Data.SiebelClient.SiebelClientFactory`, a client program can obtain an instance of the `System.Data.Common.DbCommand` class to construct a Siebel command.

```
//In this example, factory is an instance of SiebelClientFactory
DbCommand command = factory.CreateCommand();
```

Alternatively, the following approach can be used to create a command:

```
//Here connection is an instance of SiebelConnection
SiebelCommand cmd = (SiebelCommand) connection.CreateCommand();
cmd.CommandText = "SELECT [Name] as MyName, [City], [Country] from Account.Account WHERE Name LIKE '3Com*' OPTION 'ViewMode 1'";
```

The `SiebelCommand` class inherits from `DbCommand`. It exists in the namespace `Microsoft.Data.SiebelClient`.

Supported Properties

The **SiebelCommand** class supports the following `DbCommand` *protected* properties:

Name	Get/Set	Description
DbConnection	Get and Set	This should contain the underlying <code>DbConnection</code> instance from which this <code>DbCommand</code> instance is obtained.
DbParameterCollection	Get	Gets the collection of <code>DbParameter</code> objects.

`SiebelCommand` also supports the following `DbCommand` *public* properties:

Name	Get/Set	Description
CommandText	Get and Set	This contain the SQL statement that the ADO.NET client wishes to execute.
CommandType	Get and Set	Only <code>CommandType.Text</code> is supported.
Connection	Get and Set	This uses the <code>DbConnection</code> member.
Parameters	Get	This uses the <code>DbParameterCollection</code> member.
◆ Important		
The <code>SiebelCommand</code> class ignores the <code>CommandTimeout</code> , <code>DesignTimeVisible</code> , and <code>DbTransaction</code> properties.		

Supported Methods

The Data Provider for Siebel supports the following `DbCommand` *protected* methods:

Name	Description
CreateDbParameter	Creates a new <code>DbParameter</code> instance.
ExecuteDbDataReader	This executes the SELECT and EXEC commands and returns a <code>DbDataReader</code> .

`SiebelCommand` also supports the following `DbCommand` *public* methods:

Name	Description
CreateParameter	Creates a new <code>DbParameter</code> instance through <code>CreateDbParameter()</code> .
ExecuteReader	Executes <code>CommandText</code> against the <code>Connection</code> and returns <code>DbDataReader</code> through <code>ExecuteDbDataReader()</code> .
Prepare	This parses the <code>CommandText</code> and builds the SQL command parse tree.

See Also

Other Resources

[Extending ADO.NET Interfaces](#)

SiebelParameter

The Data Provider for Siebel provides a `DbParameter` implementation to enable an ADO.NET client to specify parameters for a particular command. Using an instance of the `System.Data.Common.DbCommand` class of the Data Provider for Siebel, a client program can obtain an instance of the `System.Data.Common.DbParameter` class.

```
//In this example, command is an instance of DbCommand
DbParameter param = command.CreateParameter();
```

Alternatively, the following approach can be used:

```
//Here command is an instance of SiebelCommand
SiebelParameter param = (SiebelParameter) command.CreateParameter();
param.ParameterName = "@Time";
```

The `SiebelParameter` class inherits from `DbParameter`. It exists in the namespace `Microsoft.Data.SiebelClient`.

Supported Properties

The `SiebelParameter` class supports the following `DbParameter` *public* properties:

Name	Get/Set	Description
DbType	Get and Set	Data type of the parameter. See Basic Siebel Data Types .
Direction	Get and Set	The following values are supported: <ul style="list-style-type: none"> • <code>ParameterDirection.Input</code> • <code>ParameterDirection.Output</code> • <code>ParameterDirection.InputOutput</code>
IsNullable	Get and Set	The property is not supported, and will throw an exception if called.
ParameterName	Get and Set	The Data Provider for Siebel supports this property for an ADO.NET client to specify the parameter name.
Value	Get and Set	The Data Provider for Siebel represents parameter values as strings.

Supported Data Types

The following table summarizes the simple Siebel field types that Data Provider for Siebel supports. For more detailed coverage, see [Basic Siebel Data Types](#).

Siebel Field Type	.NET Type	XML Schema Type
DTYPE_BOOL	Boolean	xsd:boolean
DTYPE_CURRENCY	Decimal	xsd:decimal
DTYPE_DATE	DateTime	xsd:dateTime
DTYPE_DATETIME	DateTime	xsd:dateTime

DTYPE_UTCDATETIME	DateTime	xsd:dateTime
DTYPE_ID	String	xsd:string
DTYPE_INTEGER	Integer	xsd:int
DTYPE_NOTE	String	xsd:string
DTYPE_NUMBER	Decimal	xsd:decimal
DTYPE_PHONE	String	xsd:string
DTYPE_TEXT	String	xsd:string
DTYPE_TIME	DateTime	xsd:dateTime

Supported Methods

The `SiebelParameter` class does not override any special methods in `DbParameter`.

See Also

Other Resources

[Extending ADO.NET Interfaces](#)

SiebelConnectionStringBuilder

The Data Provider for Siebel provides a `DbConnectionStringBuilder` implementation to enable SQL Server Integration Services (SSIS) and Visual Studio explorer tools environments to get the connection properties of the Data Provider for Siebel and display them in the form of a GUI driven by the PropertyGrid.

To create a `SiebelConnectionStringBuilder`, use the following approach:

```
//In this example, factory is an instance of SiebelClientFactory
SiebelConnectionStringBuilder bldr = (SiebelConnectionStringBuilder)factory.CreateConnectio
nStringBuilder();
bldr.ConnectionString = connstr;
```

Or, simply:

```
SiebelConnectionStringBuilder bldr = new SiebelConnectionStringBuilder();
```

To see the keys and values supported as part of the `DbConnectionStringBuilder`, see [The Data Provider for Siebel Connection String](#).

See Also

Other Resources

[Extending ADO.NET Interfaces](#)

DbDataReader

The Data Provider for Siebel provides a `DbDataReader` leveraging the XML Data Reader. This provides the consumer of the Siebel data source the ability to read a forward-only stream of rows.

Supported Properties

The Data Provider for Siebel supports the following `DbDataReader` properties.

Name	Get/Set	Description
HasRows	Get	This property is not supported, and will throw an exception if accessed.
IsClosed	Get	Gets a value indicating whether the <code>DbDataReader</code> is closed.
RecordsAffected	Get	Gets a value indicating whether the <code>DbDataReader</code> contains one or more rows.
Item(Int32)	Get	Gets the value of the specified column as an instance of <code>Object</code> . Use the ordinal number for the desired column when invoking this indexer.
Item(String)	Get	Gets the value of the specified column as an instance of <code>Object</code> . Use the name of the desired column when invoking this indexer.

Supported Methods

The Data Provider for Siebel supports the following `DbDataReader` methods.

Name	Description
GetSchemaTable	<p>Returns a <code>DataTable</code> that describes the column metadata of the <code>DbDataReader</code>. The schema column attributes supported by the Data Provider for Siebel are:</p> <ul style="list-style-type: none"> • ColumnName • ColumnOrdinal • .NET DataType • Length • Precision (if available) • Scale (if available) • AllowDBNull • LocalName • Extended LocalName • Namespace
GetString	Gets the value of the specified column as an instance of <code>String</code> .
GetValue	Gets the value of the specified column as an instance of <code>String</code> .

isDBNull	Gets a value that indicates whether the column contains nonexistent or missing values.
NextResult	The Siebel Data Provider always returns a single result set; hence this call fully exhausts the current result set before returning false .
Read	Advances the reader to the next record in a result set. It returns true if it succeeds, and false if the reader has no more records left.
Close	Closes the <code>DbDataReader</code> object.  Caution When you are done using the <code>DbDataReader</code> object, you must close it, in order to free up the Siebel COM library objects. Otherwise, the client application's memory and handle usage will go up.

See Also

Other Resources

[Extending ADO.NET Interfaces](#)

The Data Provider for Siebel Connection String

The .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel) uses the Siebel adapter to access the Siebel system. The Siebel adapter in turn uses the Siebel COM Data Control library to access the Siebel system. The Siebel COM Data Control comes bundled with the Siebel Web client.

To establish connectivity to a Siebel system, ADO.NET clients must specify the Siebel connection properties that are encoded into a database connection string. This is required because the Data Provider for Siebel implements **DbConnection** to access the underlying Siebel adapter binding.

The connection string to connect to a Siebel system using the Data Provider for Siebel contains the following properties.

Property	Description
Password	The password for the user on the Siebel system; this value is case-sensitive.  Note The Data Provider for Siebel preserves the case of the value that you enter for the password when it opens a connection on the Siebel system.
Username	The user name on the Siebel system; this value is case-sensitive.  Note The Data Provider for Siebel preserves the case of the value that you enter for the user name when it opens a connection on the Siebel system.
Compression	The compression algorithm to use between the Data Provider for Siebel and the Siebel system. Supported values are none or zlib . This parameter is optional. If it is not specified, the Siebel system supplies a default value (zlib).
Encryption	The type of encryption to use between the Data Provider for Siebel and the Siebel system. Supported values are none , mscrypto , or rsa . This parameter is optional. If it is not specified, the Siebel system supplies a default value (none).
Language	The language of the object manager. An example value is enu . This parameter is required.
SiebelEnterpriseServer	The name of the Siebel Enterprise Server. This parameter is required.
SiebelGateway	Consists of the Siebel server IP and port. For example, Siebel_Server:1234.
SiebelObjectManager	The name of the Siebel object manager on the enterprise server. This parameter is required.
SiebelRepository	The Siebel repository. Required if more than one repository exists on the server; otherwise, optional.  Note If more than one repository exists on the server, you must specify a target repository in the SiebelRepository parameter.
SiebelServer	The Siebel server. Required for all Siebel 7.5 server connections; otherwise, do not set this parameter.
Transport	The transport; only tcPIP is supported. This parameter is optional. If it is not specified, the Siebel system supplies a default value (tcPIP).

For example:

```
Username=YourUserName;Password=YourPassword;SiebelGateway=Siebel_Server:1234;SiebelObjectMa  
nager=obj_mgr;SiebelEnterpriseServer=ent_server;Language=enu;SiebelRepository=siebel_rep
```

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

Syntax for a SELECT Statement

Using the Data Provider for Siebel, ADO.NET clients can perform a SELECT query on Siebel business components by specifying a WHERE clause that represents a valid Siebel search specification. The syntax for the SELECT statement is:

```
SELECT
<column name 1> AS <column alias 1>,
<column name 2> AS <column alias 2>,
...
FROM
<Business object name>.<Business component name> AS <table alias>
WHERE
<filter condition>
OPTION
'ViewMode <value>'
```

In the above syntax, the ViewMode option corresponds to the Siebel system View Modes, which is a filtering mechanism to restrict the set of records that match the query. For the allowed set of values, see the Siebel documentation.

Note

If the field names in the WHERE clause contain special characters or empty spaces, make sure you always enclose the field names within square brackets.

Note

In SELECT queries containing alias names with special characters, make sure you include the alias names within square brackets.

Note

The Data Provider for Siebel supports alias names for tables in the SELECT clause but not in the WHERE clause.

Searching and Sorting Data Using the Data Provider for Siebel

The Data Provider for Siebel supports a filter condition in SQL statements based on the search specifications supported by the Siebel system.

The rules for the search specification are:

- Standard comparison operators must be used to compare a field to a constant, or one field to another field. These include =, !=, >, <, >=, and <=.

Example: [Revenue] > 5000

- String constants must be enclosed in double quotation marks, and the string values must be case-sensitive.

Example: [Type] != "COST LIST"

- The logical operators AND, OR, and NOT must be used to negate or combine expressions. Case sensitivity is ignored in these operators; for example, "and" is the same as "AND".

Example: [Competitor] IS NOT NULL and [Competitor] != "N"

- A field name in a search specification must be enclosed in square brackets.

Example: [Conflict Id] = 0

- The LIKE operator may be used to create text string comparison expressions in which a field is compared to a constant, or a field to another field and a match on only the first several characters is required. The wildcard characters "*" and "?" must be used to indicate any number of characters, and a single character, respectively.
- ADO.NET clients can specify original Siebel business objects, business components, and business component field names. These names must be enclosed in square brackets if they contain any special characters or white space. Examples of queries that are supported are:

```
SELECT [Name], [Postal Code] FROM Account.Account where [Postal Code] != '11065'  
SELECT [Name], [Postal Code], Id From Account.Account where [Postal Code] != '60626' O  
rder BY Id ASC, Name DESC  
SELECT * FROM [Admin Price List].[Price Book Items]
```

The Data Provider for Siebel supports sort specifications in SQL statements based on the sort specification supported by Siebel. The rules for the sort specification are:

- Use commas to separate field names in a sort specification; for instance, Name, Location
- To indicate that a field in the list sorts in descending order, include (DESC) after the field name, as in "Start Date (DESC)." If no sort order is specified, ascending order is used. To explicitly specify ascending order, use the keyword (ASC).
- The sort specification expression must be 255 characters or less.

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

Syntax for an EXEC Statement

Using the Data Provider for Siebel, ADO.NET clients can also perform an EXEC operation on the Siebel adapter. The syntax for the EXEC statement is:

```
EXEC
<Business Service name>.<Business Service method>
<value 1..n>,
@parameter 1..n [OUTPUT],
@parameter 1..n = <value>
```

In the preceding syntax, <value 1..n> represents a set of unnamed parameters. These are hard-coded values. They usually represent IN parameters. They can also represent INOUT parameters. However, if a hard-coded value is used for an INOUT parameter, the output value associated with that parameter cannot be retrieved after the EXEC statement is executed.

The @parameter 1..n syntax represents a set of named parameters, which can be IN, INOUT, or OUT parameters. The output parameters must be followed by the **OUTPUT** keyword.

Note

The **OUTPUT** keyword must only be used with OUT parameters and not with INOUT parameters.

To specify parameter values inline, use the @parameter 1..n = <value> syntax.

All parameters must be comma-separated.

The following are examples of EXEC statements:

```
EXEC ExtractDataService.Echo @In, @InOut, @Out OUTPUT
EXEC ExtractDataService.Echo 'InputValue', @InOut, @Out OUTPUT
EXEC ExtractDataService.Echo @InOut, @Out OUTPUT, @In='InputValue'
EXEC ExtractDataService.Echo 'InputValue', @Out OUTPUT, @InOut='InputValue'
```

Note

Every parameter name (like @In in the preceding example) must match the corresponding argument name in the Siebel Business Service method.

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

Performing a SELECT Query on Business Components

This section demonstrates how to select data from a Siebel business component using the .NET Framework Data Provider for Siebel eBusiness Applications.

Selecting Data from a Siebel Business Component

This section demonstrates how to select data from the "Account" business component in the Siebel repository.

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data.Common;
using Microsoft.Adapters.SiebelDbProvider;

namespace SiebelADO
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                SiebelProviderFactory factory = SiebelProviderFactory.Instance;
                DbConnection connection = factory.CreateConnection();
                connection.ConnectionString = "Username=SADMIN;Password=SADMIN;ServiceUri=172.23.115.223:2321;SiebelObjectManager=SSEObjMgr;SiebelEnterpriseServer=ent771;Language=enu;SiebelRepository=Siebel Repository";

                connection.Open();
                DbCommand command = connection.CreateCommand();

                //Here is a list of sample commands. The second command shows how the Siebel ViewMode can be specified in the query.
                //command.CommandText = "SELECT * from Account.Account where [Name] LIKE '3Com'";
                //command.CommandText = "Select [Name] from [Account].[Account] where [Name] LIKE '3Com' OPTION 'ViewMode 1'";
                command.CommandText = "SELECT [Name], [Postal Code], Id From Account.Account where [Postal Code] != '60626' Order BY Id ASC, Name DESC";

                DbDataReader dbReader = command.ExecuteReader();

                while (dbReader.Read())
                {
                    for (int i = 0; i < dbReader.FieldCount; i++)
                    {
                        string name = dbReader.GetName(i);
                        string val = dbReader[i].ToString();
                        Console.WriteLine(name + "\t:" + val);
                        Console.WriteLine("=====");
                    }
                }
                Console.WriteLine("Press any key...");
                Console.ReadLine();
            }
            catch (Exception exp) { Console.WriteLine(exp.Message); }
        }
    }
}
```

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

[Performing an EXECUTE Operation on Business Services](#)

Performing an EXECUTE Operation on Business Services

This section demonstrates how to execute an operation on a Siebel business service using the .NET Framework Data Provider for Siebel eBusiness Applications.

Executing a Siebel Business Service

This section demonstrates how to execute operations on a business service in the Siebel repository.

```

using System;
using System.Collections.Generic;
using System.Text;
using System.Data.Common;
using System.Data;
using Microsoft.Adapters.SiebelDbProvider;

namespace SiebelADOBS
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                SiebelProviderFactory factory = SiebelProviderFactory.Instance;
                DbConnection connection = factory.CreateConnection();
                connection.ConnectionString = "Username=SADMIN;Password=SADMIN;ServiceUri=172.23.11
5.223:2321;SiebelObjectManager=SSEObjMgr;SiebelEnterpriseServer=ent771;Language=enu;SiebelR
epository=Siebel Repository";

                connection.Open();
                DbCommand command = connection.CreateCommand();
                command.CommandText = "EXEC ExtractDataService.Echo @In, @InOut, @Out OUTPUT";

                //Add @In
                DbParameter param1 = command.CreateParameter();
                param1.ParameterName = "@In";
                param1.Direction = ParameterDirection.Input;
                param1.Value = "SomethingElse";
                command.Parameters.Add(param1);

                //Add @InOut
                DbParameter param2 = command.CreateParameter();
                param2.ParameterName = "@InOut";
                param2.Direction = ParameterDirection.InputOutput;
                command.Parameters.Add(param2);

                //Add @Out
                DbParameter outParam = command.CreateParameter();
                outParam.ParameterName = "@Out";
                outParam.Direction = ParameterDirection.Output;
                command.Parameters.Add(outParam);

                DbDataReader dbReader = command.ExecuteReader();

                Console.WriteLine("Param2: " + param2.Value);
                Console.WriteLine("OutParam: " + outParam.Value);

                Console.WriteLine("Press any key...");
                Console.ReadLine();
            }
            catch (Exception exp) { Console.WriteLine(exp.Message); }
        }
    }
}

```

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)
[Performing a SELECT Query on Business Components](#)

Using the Data Provider for Siebel with SSIS

You can use the .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel) along with SQL Server Integration Services (SSIS) to import data from a Siebel system into SQL Server database tables, flat files, or other compatible destination types. Specifically, you can create an SSIS package that can be executed to import this data.

To import data into the SQL Server database, use the SQL Server Import and Export Wizard, and provide a SELECT query to specify the data to be imported. The query must conform to the semantics supported by the Data Provider for Siebel. For more information about the grammar for a SELECT query for the Data Provider for Siebel, see [Syntax for a SELECT Statement](#).

Note

The Data Provider for Siebel does not support using an EXEC statement with SSIS.

You can start the SQL Server Import and Export wizard either from the SQL Server Management Studio or from an Integration Service project in Visual Studio. This section provides instructions on running the wizard from both the SQL Server Management Studio and Visual Studio interfaces.

In This Section

- [Importing Siebel Data by Using SQL Server Management Studio](#)
- [Importing Siebel Data by Using Visual Studio](#)

See Also

Concepts

[Using the .NET Framework Data Provider for Siebel eBusiness Applications](#)

Importing Siebel Data by Using SQL Server Management Studio

This section provides information about how to use SQL Server Management Studio to import data from a Siebel system into a SQL Server database. It also provides instructions on how to create and execute an SSIS package to import this data.

Prerequisites

Before performing the procedures provided in this topic, make sure:

- The Data Provider for Siebel is installed on the computer.
- SQL Server Business Intelligence Development Studio is installed on the computer.

Importing Data by Using SQL Server Management Studio

Perform the following steps to import data from Siebel system using Data Provider for Siebel with SQL Server Management Studio.

To import data by using SQL Server Management Studio

1. Start the SQL Server Management Studio.
2. In the **Connect to Server** dialog box, specify the values to connect to a SQL Server database, and then click **Connect**. Microsoft SQL Server Management Studio opens.
3. In Object Explorer, expand the SQL Server name, expand **Databases**, and right-click the database into which you will be exporting the tables from the Siebel system. From the context menu, point to **Tasks**, and then click **Import Data**. This starts the SQL Server Import and Export Wizard.
4. Read the information on the Welcome screen, and then click **Next**.
5. In the **Choose a Data Source** dialog box, from the **Data Source** drop-down list, select **.NET Framework Data Provider for Siebel eBusiness Applications**. Specify values for the different connection properties for the Data Provider for Siebel connection string. For more information about the connection string properties, see [The Data Provider for Siebel Connection String](#).

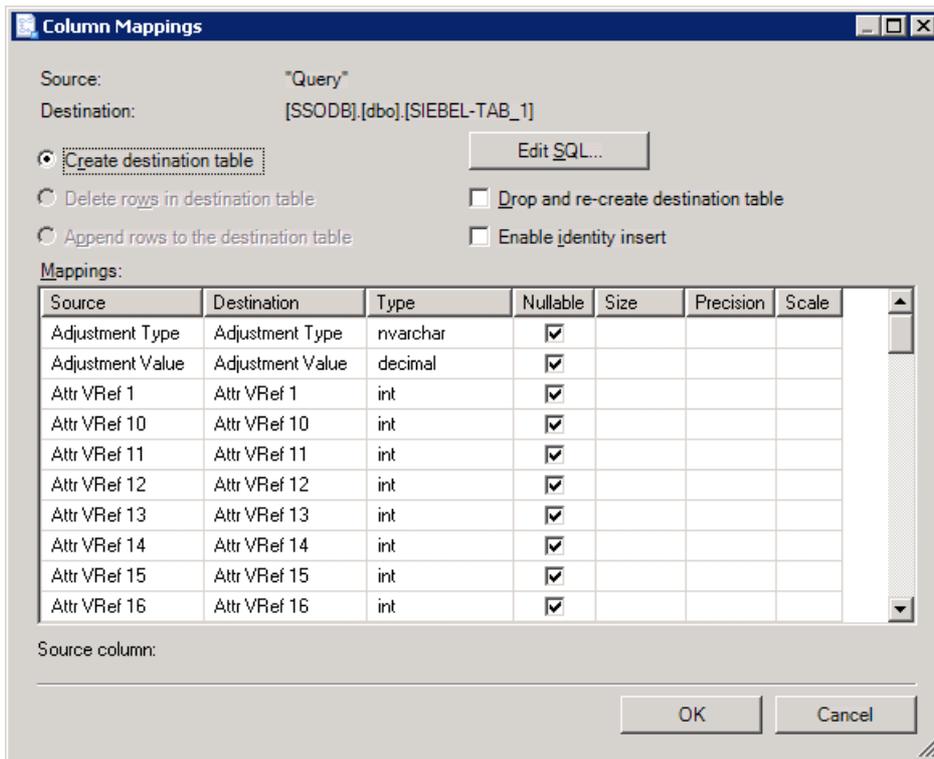
Click **Next**.

6. In the **Choose a Destination** dialog box:
 - a. From the **Destination** drop-down list, select **SQL Native Client**.
 - b. From the **Server name** drop-down list, select a SQL Server name.
 - c. Select an authentication mode.
 - d. From the **Database** drop-down list, select the database to which you want to import the Siebel table.
 - e. Click **Next**.
7. In the **Specify Table Copy or Query** dialog box, choose the **Write a query to specify the data to transfer** option.
8. In the **Provide a Source Query** dialog box, specify a SELECT query to filter the data to be imported into the SQL Server. For more information about the grammar for a SELECT query for the Data Provider for Siebel, see [Syntax for a SELECT Statement](#).

Click the **Parse** button to validate the query, click **OK** in the pop-up dialog box, and then click **Next**.

9. In the **Select Source Tables and Views** dialog box, select the check box against the source and destination tables. The source is the query you specified to retrieve data from Siebel. The destination is the table that will be created in the SQL Server database.

10. The wizard creates a default mapping between the source and destination table fields. However, you can change the mappings according to your requirement. To change the field mappings, click **Edit Mappings**.



11. In the **Column Mappings** dialog box, you can:

- Change the names of columns in the destination table.
- Ignore certain columns in the destination table.
- Change the data type for fields in destination table.
- Change other field attributes such as nullable, size, precision, and scale.

When you are finished, click **OK**.

12. In the **Select Source Tables and Views** dialog box, click **Next**.

13. In the **Save and Execute Package** dialog box:

- Select the **Execute immediately** check box to execute the query.
- Select the **Save SSIS Package** check box to save the query as a package and execute it later. If you chose to save the package, you must also specify whether you want to save the package in the SQL Server or the file system.
- From the **Package protection level** drop-down list, select a protection level for the package and specify credentials where required.
- Click **Next**.

If you chose to save the package, proceed to the next step. Otherwise, skip to step 15.

14. In the **Save SSIS Package** dialog box, specify the following:

- The name for the package
- The description for the package

- If you chose to save the package to a SQL Server, select a SQL Server from the **Server name** drop-down list.
- If you chose to save the package to the file system, specify the name and location of the file in the **File name** text box.

When you are finished, click **Next**.

15. In the **Complete the Wizard** dialog box, review the summary of actions that the wizard will perform, and then click **Finish**.
16. In the **Performing Operations** dialog box, the wizard starts executing tasks to import the information from Siebel into a SQL Server database table. The status for each task is displayed in the wizard.
17. After all the tasks are successfully executed, click **Close**. If a task fails, see the corresponding error message, fix the issue, and rerun the wizard.

Running the SSIS Package

If you chose to save the SSIS package, you can run it to retrieve the most recent information from the Siebel system. This section provides information about how to run the package if you chose to save it to the file system.

To run the package from Windows Explorer

1. From **Windows Explorer**, navigate to the location where you saved the package, and double-click the package.
2. In the **Execute Package Utility** dialog box, click **Execute**. The **Package Execution Progress** dialog box displays the progress of the different tasks.
3. After all the tasks are successfully executed, click **Close**.
4. In the **Execute Package Utility** dialog box, click **Close**.

For more information about running packages, see "Running Packages" at <http://go.microsoft.com/fwlink/?LinkId=94972>. For any other information related to SSIS packages, see "Package How-to Topics (SSIS)" at <http://go.microsoft.com/fwlink/?LinkId=94973>.

Verifying the Results

After executing the package, you must verify the results by going to the SQL Server database to which the Siebel data is imported. Executing the package should have created a table in the destination database. This table should be populated with the values from the Siebel table.

See Also

Tasks

[Using the Data Provider for Siebel with SSIS](#)

Importing Siebel Data by Using Visual Studio

This section provides information about how to use Microsoft Visual Studio to import data from a Siebel system into a SQL Server database. It also provides instructions on how to create and execute an SSIS package to import this data.

Prerequisites

Before performing the procedures provided in this topic, make sure:

- The Data Provider for Siebel is installed on the computer.
- Microsoft Visual Studio 2010 is installed on the computer.

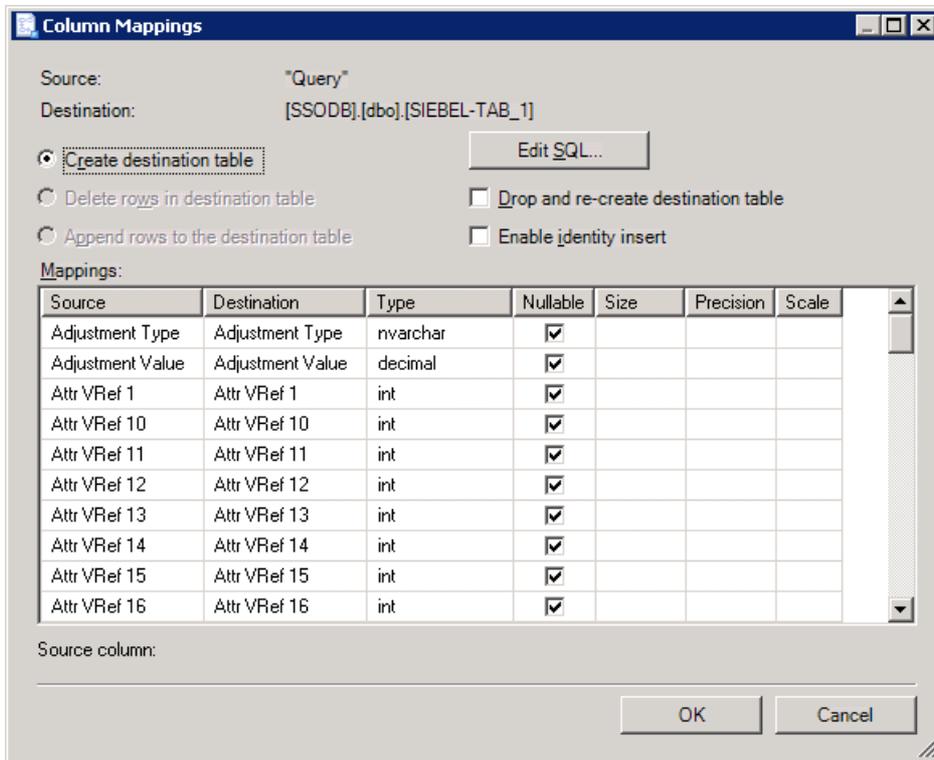
Importing Data by Using Visual Studio

Perform the following steps to import data using Data Provider for Siebel in Visual Studio.

To import data by using Visual Studio

1. Start Visual Studio and create an Integration Service project.
2. From the **Project** menu, select **SSIS Import and Export Wizard**. This starts the SQL Server Import and Export Wizard.
3. Read the information on the Welcome screen, and then click **Next**.
4. In the **Choose a Data Source** dialog box, from the **Data Source** drop-down list select **.NET Framework Data Provider for Siebel eBusiness Applications**. Specify values for the different connection properties for the Data Provider for Siebel connection string. For more information about the connection string properties, see [The Data Provider for Siebel Connection String](#).
Click **Next**.
5. In the **Choose a Destination** dialog box:
 - a. From the **Destination** drop-down list, select **SQL Native Client**.
 - b. From the **Server name** drop-down list, select a SQL Server name.
 - c. Select an authentication mode.
 - d. From the **Database** drop-down list, select the database to which you want to import the Siebel table.
 - e. Click **Next**.
6. In the **Specify Table Copy or Query** dialog box, choose the **Write a query to specify the data to transfer** option.
7. In the **Provide a Source Query** dialog box, specify a SELECT query to filter the data to be imported into the SQL Server. For more information about the grammar for a SELECT query for the Data Provider for Siebel, see [Syntax for a SELECT Statement](#).
8. To validate the query, click the **Parse** button, click **OK** in the pop-up dialog box, and then click **Next**.
9. In the **Select Source Tables and Views** dialog box, select the check box against the source and destination tables. The source is the query you specified to retrieve data from Siebel. The destination will be the table that will be created in the SQL Server database.
10. The wizard creates a default mapping between the source and destination table fields. However, you can change the mappings according to your requirement. To change the field mappings, click **Edit Mappings**.
11. In the **Column Mappings** dialog box, you can:
 - Change the names of columns in the destination table.

- Ignore certain columns in the destination table.
- Change the data type for fields in destination table.
- Change other field attributes such as nullable, size, precision, and scale.
- Click **OK**.



12. In the **Select Source Tables and Views** dialog box, click **Next**.
13. In the **Complete the Wizard** dialog box, review the summary of actions that the wizard will perform, and then click **Finish**.
14. In the **Performing Operations** dialog box, the wizard starts executing tasks to import the information from Siebel into a SQL Server database table. The status for each task is displayed in the wizard.
15. After all the tasks are successfully executed, click **Close**. If a task fails, see the corresponding error message, fix the issue, and rerun the wizard.
16. The wizard adds an SSIS package to your Integration Service project. Save the Integration Service project.

Running the SSIS Package

Once the package is created within an Integration Service project, you can execute it to import data from a Siebel system into a SQL Server database. Perform the following steps to import Siebel data by executing the package.

To run the package from Visual Studio

1. Navigate to the SSIS package in Solution Explorer.
2. Right-click the package name, and then select **Execute Package**.

For more information about running packages, see "Running Packages" at <http://go.microsoft.com/fwlink/?LinkId=94972>. For any other information related to SSIS packages, see "Package How-to Topics (SSIS)" at <http://go.microsoft.com/fwlink/?LinkId=94973>.

Verifying the Results

After executing the package, you must verify the results by logging on to the SQL Server and navigating to the database to which the Siebel data is imported. Executing the package should have created a table in the destination database. This table should be populated with the values from the Siebel table.

See Also

Tasks

[Using the Data Provider for Siebel with SSIS](#)

Using the Siebel Adapter with Microsoft Office SharePoint Server

Most line-of-business (LOB) applications have proprietary interfaces that are accessible to relatively few people with the required subject matter expertise. However, because information workers need relevant information at all levels in an organization, the subject matter experts have to export required data into more familiar applications, like Microsoft Excel, for information workers to consume the information.

The new collection of Microsoft Office client applications and features bridges this gap by bringing backend business data into the most common interface for information workers, Microsoft Office. Microsoft Office client applications seamlessly present data from LOB applications through interfaces familiar to information workers. One example is the Business Data Catalog in Microsoft Office SharePoint Server.

In This Section

- [About Business Data Catalog](#)
- [Using the Siebel Adapter to Integrate the Siebel System with the Business Data Catalog and Microsoft Office SharePoint Server](#)
- [Considerations While Using the Siebel Adapter with Microsoft Office SharePoint Server](#)
- [How Do I Use a Custom Web Part?](#)

About Business Data Catalog

The Business Data Catalog is a new integration component in Microsoft Office SharePoint Server 2007 and later. The Business Data Catalog is a shared service and surfaces business data from backend applications via databases and Web services, without the need for writing any code. It is a metadata repository that defines business entities like customers, invoices, and purchase orders. Once defined, clients can use these entities.

The Business Data Catalog provides a Business Data Catalog Definition Editor to help you author application definition files for the Business Data Catalog in Microsoft Office SharePoint Server 2007 and later. Using the Business Data Catalog Definition Editor, you can create and edit metadata for business applications.

For more information about the Business Data Catalog, see "Business Data Catalog" at <http://go.microsoft.com/fwlink/?LinkId=103882>.

See Also

Other Resources

[Using the Siebel Adapter with Microsoft Office SharePoint Server](#)

Using the Siebel Adapter to Integrate the Siebel System with the Business Data Catalog and Microsoft Office SharePoint Server

The Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK includes the WCF LOB Adapter Service Development Wizard, which generates a WCF service for specific LOB artifacts. This WCF service is hosted in a hosting environment such as Microsoft Internet Information Services (IIS). The Business Data Catalog Definition Editor uses the URL where the WCF service is hosted to get the Web Services Description Language (WSDL) for the WCF service. Using the WSDL, the Business Data Catalog Definition Editor extracts the methods available to the WCF service. These methods can be used to establish entities and the association between the entities.

The Business Data Catalog Definition Editor helps you create an application definition file (an XML file) that Microsoft Office SharePoint Server can consume. Once the application definition file is imported to Microsoft Office SharePoint Server, you can create Web Parts to present the information from enterprise applications. For more information, see "Creating Web Parts" in [Step 3: Create a SharePoint Application to Retrieve Data from Siebel](#) in [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#).

Tutorial

To demonstrate how to use the Siebel adapter with Microsoft Office SharePoint Server, the BizTalk Adapter Pack includes a tutorial that provides step-by-step instructions to present data from a Siebel system on a SharePoint site. See [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#).

See Also

Other Resources

[Using the Siebel Adapter with Microsoft Office SharePoint Server](#)

Considerations While Using the Siebel Adapter with Microsoft Office SharePoint Server

This topic contains information about the issues you might encounter while using the Microsoft BizTalk Adapter for Siebel eBusiness Applications with Microsoft Office SharePoint Server, along with resolutions. The issues are divided into two categories:

- General issues
- Issues involving custom Web Parts

General Issues

This section contains issues that either have no resolution or requires you to modify the application definition file for the resolution.

Issue 1: The simple type data returned by the WCF service is not displayed

Explanation: Microsoft Office SharePoint Server expects the data returned by the WCF service to be of DataSet or Collection type only. If the data returned by the WCF service is of simple type, Microsoft Office SharePoint Server does not display the data.

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server.

Issue 2: An error message is displayed if the data returned by the WCF service is NULL

Explanation: If the data returned by the WCF service is a NULL value, Microsoft Office SharePoint Server displays an error message. For example, suppose you are using the Business Data List Web Part for the **Finder** method instance, and are searching for customers in the Siebel system based on a search expression. The search expression that you specified fetches a NULL value. In this case, Microsoft Office SharePoint Server will display an error message.

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server.

Issue 3: An array of simple type returned by the WCF service is not displayed

Explanation: If the data returned by the WCF service is an array of simple type, Microsoft Office SharePoint Server does not display the data. Moreover, when you execute a method instance in Business Data Catalog Definition Editor that returns an array of simple type, the following error message is displayed: "Backend system adapter returned a structure incompatible with the corresponding metadata (MethodInstance, Parameter or TypeDescriptor)."

Resolution: No resolution. It is a known limitation with Microsoft Office SharePoint Server and Business Data Catalog Definition Editor.

Issue 4: Cannot import an application definition file that contains a complex type parameter having more than 300 fields

Explanation: Microsoft Office SharePoint Server cannot import an application definition file that has more than 300 fields in the complex type parameter returned by the WCF service, and displays an error message if you try to do so. This is due to the limitation of Microsoft Office SharePoint Server of not being able to display more than 300 fields of a complex type parameter.

Resolution: Use the Business Data Catalog Definition Editor to limit the number of fields of the complex type parameter to less than or equal to 300. Depending on your requirement, you can delete the fields of the complex type parameter in the Business Data Catalog Definition Editor that you do not require to be displayed in Microsoft Office SharePoint Server. Alternatively, you can also export the application definition file from Business Data Catalog Definition Editor with all the fields, and then modify the application definition file in a notepad or any XML authoring application to delete the fields that are not required in order to limit the number of fields to 300.

Issues Involving Custom Web Parts

This section contains issues that require the use of custom Web Parts for resolution. For detailed information about using a custom Web Part to resolve issues that might come up while working with Siebel adapter and Microsoft Office SharePoint Server, see [How Do I Use a Custom Web Part?](#).

Issue 1: Index of an enumerator is displayed instead of the value for the enum data type

Explanation: If a Business Data List or Business Data Item Web Part in Microsoft Office SharePoint Server contains data of enum type (a distinct type consisting of a set of named constants called the enumerators), the index of the enumerator is

displayed instead of its value in Microsoft Office SharePoint Server. This is because the Business Data List and Business Data Item Web Parts incorrectly print the values of the enum type data to the SharePoint portal.

Resolution: Use a custom Web Part to print the value of the enumerator instead of the index. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#). For example, you can use the following code sample in your Web part to print the correct values of enum type data on Microsoft Office SharePoint Server.

```
namespace CustomWebpart
{
    public class CustomWebPart : WebPart
    {
        private string displayText = "Hello World!";

        [WebBrowsable(true), Personalizable(true)]
        public string DisplayText
        {
            get { return displayText; }
            set { displayText = value; }
        }
        protected override void Render(System.Web.UI.HtmlTextWriter writer)
        {
            string SearchExpr = "[Address Name] LIKE \"*\";";
            object ElementType = null;

            /**Step 1: Get the required entity and method.***/

            LobSystem newSystem = ApplicationRegistry.GetLobSystems()["WebServiceLobSystem"]; // Name specified in application definition file
            LobSystemInstance newSystemInstance = newSystem.GetLobSystemInstances()["Siebel_Instance"]; // Name specified in application definition file
            Entity CategoryEntity = newSystem.GetEntities()["Siebel_Method_Name"]; // Name specified in application definition file
            Method newMethod = CategoryEntity.GetMethods()["Query"]; // Name specified in application definition file
            MethodInstance methodInstance = newMethod.GetMethodInstances()["MethodInstance0"]; // Name specified in application definition file

            /**Step 2: Get the list of input parameters.***/
            Object[] args = methodInstance.GetMethod().CreateDefaultParameterInstances(methodInstance); // Get default value of the input parameter
            Object[] ArgsInput = new Object[args.Length];

            /**Step 3: Assign them required values.***/

            //Assigning values to a complex type parameter. Index of this parameter is 3rd in args array.
            /** Complex Type Parameter is defined as follows:
            <Parameter Direction="In" Name="BusinessAddressQueryInputRecord">
            <TypeDescriptor TypeName="BDC.BusinessAddressQueryInputRecord,WebServiceLobSystem" Name="BusinessAddressQueryInputRecord">
            <TypeDescriptors>
            <TypeDescriptor TypeName="System.String, ..." Name="SearchExpr"></TypeDescriptor>
            <TypeDescriptor TypeName="System.String, ..." Name="SortSpec"></TypeDescriptor>
            <TypeDescriptor TypeName="System.String[], ..." IsCollection="true" Name="QueryFields"></TypeDescriptor>
            </TypeDescriptors>
            </TypeDescriptor>
            </Parameter>
            * We are assigning value to Parameter SearchExpr. ***/

            Assembly asm = Assembly.GetAssembly(args[2].GetType());
            Type t = asm.GetType(args[2].GetType().ToString()); // Get type of the parameter

            FieldInfo[] FI = t.GetFields();
            ElementType = Activator.CreateInstance(t);
        }
    }
}
```



```
        ElementType.GetType().GetFields()[f].SetValue(ElementType, (Object)"ElementValue");
    }
    ElementArray.SetValue(ElementType, ind);
}

ArgsInput[i] = (object)ElementArray; // As shown in sample, ArgsInput is fed as
input while executing Method Instance
```

Issue 3: Limitation with specifying NULL values to complex type parameters

Explanation: If you do not specify any value for a complex type parameter from a Web Part in Microsoft Office SharePoint Server, NULL should be passed on as the value for the complex type parameter to the WCF service. However, a non-NULL value is passed for the complex type parameter, and NULL is passed for its children elements (of simple type). This causes a mismatch between the expected message schema and the message schema that is passed on to the WCF service. As a result, the Siebel adapter might display an error message.

Note

To find out the default value of a complex type parameter when no value is passed from a Web Part in Microsoft Office SharePoint Server, use step 2 in the code sample mentioned in "Issue 1: Index of an enumerator is displayed instead of the value for the enum data type."

Resolution: Use a custom Web Part to assign a NULL value to the complex type parameter when no value is specified from a Web Part in Microsoft Office SharePoint Server. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#)

Issue 4: Limitation with displaying a single record in Microsoft Office SharePoint Server based on multiple values

Explanation: If you want to display a single record in Microsoft Office SharePoint Server based on multiple values (input parameters) from a Siebel system, you cannot use any of the three Web Parts (Business Data List, Business Data Item, and Business Data Related List) specified in [Step 3: Create a SharePoint Application to Retrieve Data from Siebel](#) in [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#).

Resolution: You must use a custom Web Part to do this. For information about using a custom Web Part, see [How Do I Use a Custom Web Part?](#) For example, in step 3 in "Issue 1: Index of an enumerator is displayed instead of the value for the enum data type" you can modify the code to provide values for more than one parameter instead of providing input to a single business component parameter.

See Also

Other Resources

[Using the Siebel Adapter with Microsoft Office SharePoint Server](#)

How Do I Use a Custom Web Part?

This section provides information about using a custom Web Part with Microsoft Office SharePoint Server. To use a custom Web Part, you must do the following:

1. Create a custom Web Part
2. Deploy the custom Web Part to a SharePoint portal
3. Configure the SharePoint portal to use the custom Web Part

Before You Begin

Before you create a custom Web Part:

- Publish the Siebel artifacts as a WCF service. For more information, see [Step 1: Publish the Siebel Business Component Operations as a WCF Service](#) in [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#).
- Create an application definition file for the Siebel artifacts using the Business Data Catalog in Microsoft Office SharePoint Server. For more information, see [Step 2: Create an Application Definition File for Siebel Business Component Operations](#) in [Tutorial 1: Presenting Data From a Siebel System on a SharePoint Site](#).

Step 1: Create a custom Web Part

To create a custom Web Part using Visual Studio, do the following:

1. Start Visual Studio 2010, and then create a project.
2. In the **New Project** dialog box, from the **Project types** pane, select **Visual C#**. From the **Templates** pane, select **Class Library**.
3. Specify a name and location for the solution. For this topic, specify **CustomWebPart** in the **Name** and **Solution Name** boxes. Specify a location, and then click **OK**.
4. Add a reference to the System.Web component into the project. Right-click the project name in **Solution Explorer**, and then click **Add Reference**. In the **Add Reference** dialog box, select **System.Web** in the **.NET** tab, and then click **OK**. The System.Web component contains the required namespace of System.Web.UI.WebControls.WebParts.
5. Add the required code based on your issue in the project. For the code sample that is relevant to a certain issue, see "Issues Involving Custom Web Parts" in [Considerations While Using the Siebel Adapter with Microsoft Office SharePoint Server](#).
6. Build the project. On successful build of the project, a .dll file, CustomWebPart.dll, will be generated in the *<project folder>/bin/Debug* folder.

Step 2: Deploy the custom Web Part to a SharePoint Portal

You must do the following to make the CustomWebPart.dll file (custom Web Part) that is created in "Step 1: Create a custom Web Part" of this topic usable on the SharePoint portal:

- **Copy the CustomWebPart.dll file to the bin folder of the SharePoint Portal:** Microsoft Office SharePoint Server creates portals under the *<root drive>:\Inetpub\wwwroot\wss\VirtualDirectories* folder. A folder is created for each portal, and can be identified with the port number. You must copy the CustomWebPart.dll file created in "Step 1: Create a custom Web Part" of this topic to the *<root drive>:\Inetpub\wwwroot\wss\VirtualDirectories\<Port_Number>\bin* folder. For example, if the port number of your SharePoint portal is 13614, you must copy the CustomWebPart.dll file to the *<root drive>:\Inetpub\wwwroot\wss\VirtualDirectories\13614\bin* folder.

Tip

Another way to find the folder location of your SharePoint portal is by using the **Internet Information Services (IIS) Manager** window (**Start > Run > inetmgr**). Locate your SharePoint portal in the **Internet Information Services (IIS) Manager** window ([*computer_name*] > Web Sites > [*Portal-Name*]), right-click, and then click **Properties** in the shortcut menu. In the properties dialog box of the SharePoint portal, click the **Home Directory** tab, and then select the **Local path** box.

- **Add the Safe Control Entry in the web.config File:** Because the CustomWebPart.dll file will be used on different computers and by multiple users, you must declare the file as "safe." To do so, open the web.config file located in the SharePoint portal folder at <root drive>:\inetpub\wwwroot\wss\VirtualDirectories\<Port_Number>. Under the <SafeControls> section of the web.config file, add the following safe control entry:

```
<SafeControl Assembly="CustomWebPart" Namespace="CustomWebPart" TypeName="*" Safe="True" />
```

Save the web.config file, and then close it.

Step 3: Configure the SharePoint Portal to use the custom Web Part

You need to add the custom Web Part to the Microsoft Office SharePoint Server Web Part Gallery, so that you can use it on your SharePoint portal. To do so:

1. Start SharePoint 3.0 Central Administration. Click **Start**, point to **All Programs**, point to **Microsoft Office Server**, and then click **SharePoint 3.0 Central Administration**.
2. In the left navigation pane, click the name of the Shared Services Provider (SSP) to which you want to add the custom Web Part.
3. On the **Shared Services Administration** page, in the upper-right corner, click **Site Actions**, and then click **Create**.
4. On the **Site Settings** page, click **Web Parts** under the **Galleries** column.
5. On the **Web Part Gallery** page, to add the custom Web Part to the gallery, click **New**. At this point the custom Web Part is not available in the **Web Part Gallery** page.
6. On the **New Web Parts** page, locate **CustomWebPart** (name of the custom Web Part) in the list, select the check box on the left, and then click **Populate Gallery** on the top of the page. This will add the **CustomWebPart** entry in the **Web Part Gallery** page.

Now you can use the custom Web Part (**CustomWebPart**) to create Web Parts in your SharePoint portal. The custom Web Part (**CustomWebPart**) will appear under the **Miscellaneous** section in the **Add Web Parts** page.

See Also

Other Resources

[Using the Siebel Adapter with Microsoft Office SharePoint Server](#)

Samples

Samples for Microsoft BizTalk Adapter for Siebel eBusiness Applications are categorized into:

- BizTalk Server samples
- WCF service model samples
- Data Provider for Siebel samples
- Migration samples

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>.

The following list contains the names and descriptions of the samples for the Siebel adapter.

BizTalk Server Samples

Sample Directory Name	Description
BusinessService	Demonstrates how to invoke a business service in Siebel using the Siebel adapter.
MVLDemo	Demonstrates how to work with multivalued links (MVLs) in Siebel using the Siebel adapter.
SiebelAccount	Demonstrates how to insert records into the Account business component in Siebel using the Siebel adapter.
SiebelAdapterIntegration Objects	Demonstrates how to invoke a business service in Siebel, which works with Integration Objects, using the Siebel adapter.
SiebelPicklist	Demonstrates how to insert values of picklist types into a Siebel business component using the Siebel adapter.

WCF Service Model Samples

Sample Directory Name	Description
AccountInsertDelete	Demonstrates how to perform Insert, Update, Delete, and Query operations on a Siebel business component. The sample inserts a record into the Account business component of the Account business object, updates the record, and finally deletes it. Before and after each operation a Query operation is performed on the business component to verify the results.
Business Services	Demonstrates how to invoke a Siebel business service method, TimeStamp, and displays the results to the console.
MVL	Demonstrates how to work with multivalued links in a Siebel business component by using the Associate, Dissociate, and child query operations. The sample displays all contacts associated with an account. It also shows how to Associate and Dissociate a contact from an account.

Data Provider for Siebel Sample

Sample Directory Name	Description
siebel ado	Demonstrates how to use the .NET Framework Data Provider for Siebel eBusiness Applications.

Migration Sample

Sample Directory Name	Description
-----------------------	-------------

Siebel_Buss_Comp	Demonstrates how to use a BizTalk project created using the previous version of the Siebel adapter and make it work with the WCF-based Siebel adapter.
------------------	--

See Also

Concepts

[Development](#)

Deployment

This section provides information about deploying the BizTalk Adapter Pack. This section also provides troubleshooting information about common installation issues and some frequently asked questions.

In This Section

- [Installing the BizTalk Adapter Pack](#)
- [Troubleshooting Your Installation](#)
- [Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Installing the BizTalk Adapter Pack

See the BizTalk Adapter Pack installation guide for installation prerequisites and complete instructions for installing the BizTalk Adapter Pack. This document is installed on your computer, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.

See Also

Concepts

[Deployment](#)

Other Resources

[Troubleshooting Your Installation](#)

Troubleshooting Your Installation

The Microsoft BizTalk Adapter Pack installation copies the product binaries on a computer and registers the bindings for each adapter. For information about troubleshooting BizTalk Adapter Pack installation issues, see [Troubleshooting Installation Issues](#).

See Also

Concepts

[Deployment](#)

Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter

The previous version of the Siebel adapter that shipped with Microsoft BizTalk Server differs from Microsoft BizTalk Adapter for Siebel eBusiness Applications in many aspects, including:

- The design-time experience of creating a BizTalk project.
- The metadata retrieval experience.
- Schemas.
- Data type mappings.
- The operations that can be performed using the adapter.
- Physical port configuration in the BizTalk Server Administration console.

This section discusses the differences between the versions.

Note

The Siebel adapter is a WCF-based adapter and can be used with BizTalk Server, the WCF service model, or the WCF channel model. The earlier version of the adapter, however, can be used only with BizTalk Server.

Therefore, this section talks about version differences in terms of using the adapter with BizTalk Server. For information about how to use the adapter with the WCF service model or WCF channel model, see

[Overview of BizTalk Adapter for Siebel eBusiness Applications](#).

In This Section

- [Differences in the Design-time Experience](#)
- [Differences in Metadata Retrieval Experience](#)
- [Differences in the Schema File Name and Namespace](#)
- [Differences in Data Type Mapping](#)
- [Differences in Operations Surfaced for the Siebel Artifacts](#)
- [Differences in Physical Port Configuration in the BizTalk Server Administration Console](#)
- [Migration Tutorials and Sample](#)

See Also

Concepts

[Deployment](#)

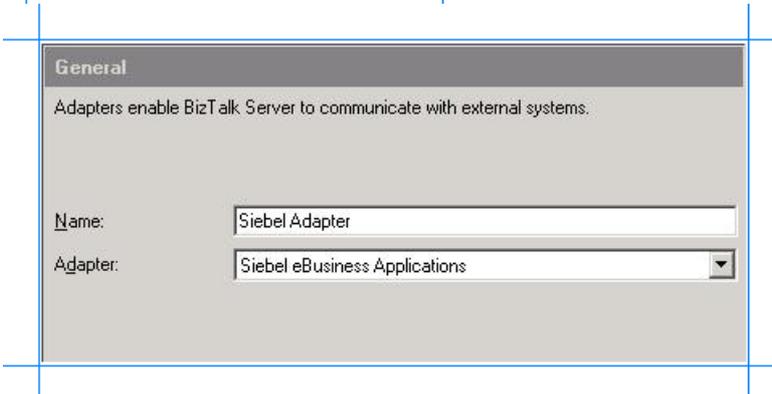
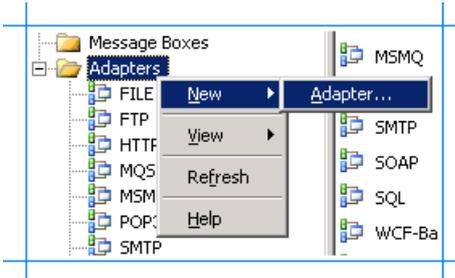
Differences in the Design-time Experience

The design-time experience for using the adapters includes creating a BizTalk project in Visual Studio and then generating metadata for the operations you want to perform on the Siebel system using the adapter. This section discusses how the design-time experience differs between the two adapter versions.

Design-time Experience for the Previous Version of the Adapter

The design-time experience using the previous version of the adapter involves the following steps:

1. **Add the Siebel adapter to the BizTalk Server Administration console.** Installing the adapter does not add the adapter to the console. You must add the adapter in the BizTalk Server Administration console under BizTalk Server Administration > BizTalk Group > Platform Settings > Adapters.

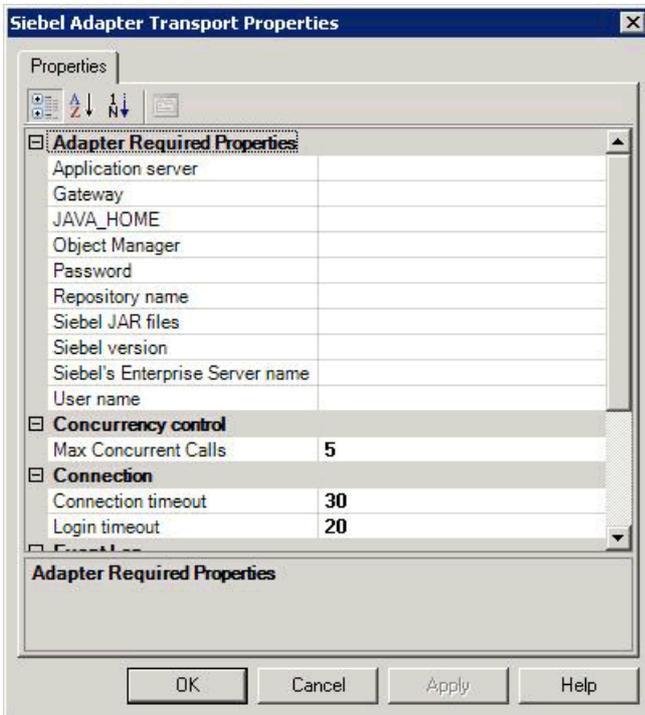


2. **Add a physical send port.** To retrieve metadata from a Siebel system, the previous version of the adapter requires you to configure a physical send port in BizTalk Server Administration console, before you create a BizTalk project.

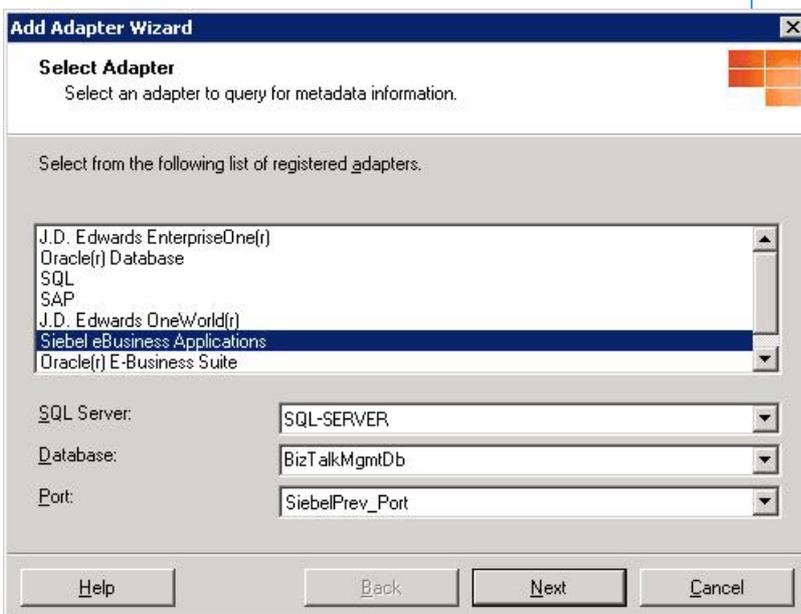
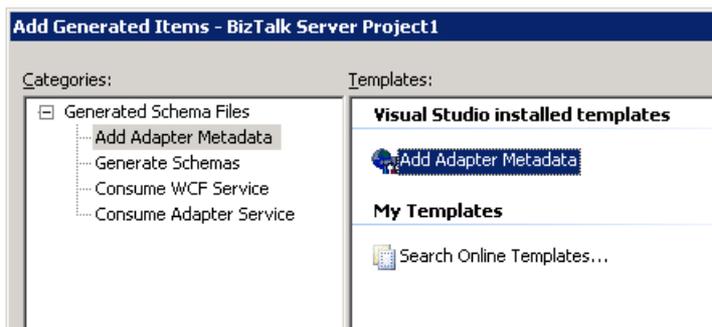
You can configure a send port under BizTalk Server Administration > BizTalk Group > BizTalk Applications > [Your Application] > Send Ports. For the send port, you must specify the connection properties to connect to the Siebel system.

The key connection properties to connect to a Siebel system using the previous version of the adapter are:

- **Without Enterprise SSO:** User name, Password, Gateway, Object Manager, repository name, Siebel Enterprise Server name, Affiliate Application (set to none).
- **With Enterprise SSO:** ESSO affiliate application configured in the SSO Administration console, all other parameters mentioned above (except user name and password).



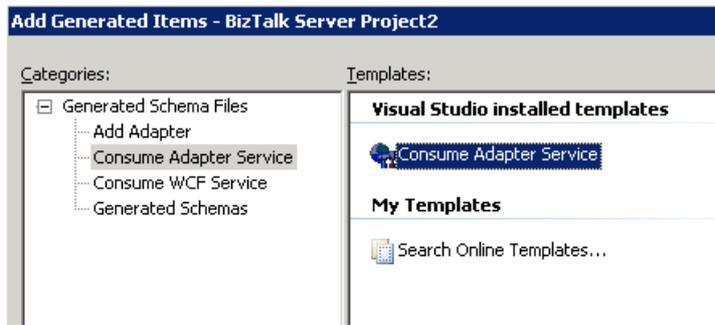
3. **Generate metadata.** Create a BizTalk project in Visual Studio. Right-click the BizTalk project and select Add > Add Generated Items > Add Adapter Metadata.



Design-time Experience for the New WCF-based Adapter

The design-time experience using the WCF-based Siebel adapter involves the following steps:

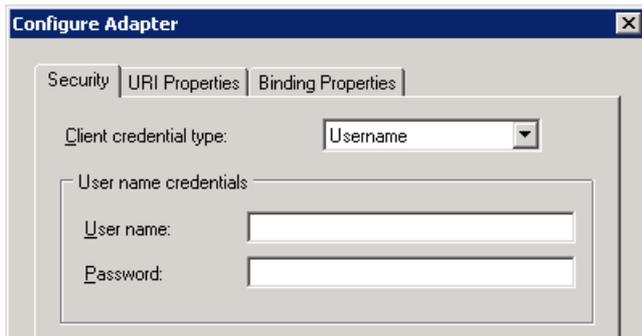
1. **Create a BizTalk project.** Create a BizTalk project in Visual Studio. Right-click the BizTalk project and select Add > Add Generated Items > Consume Adapter Service.



Note

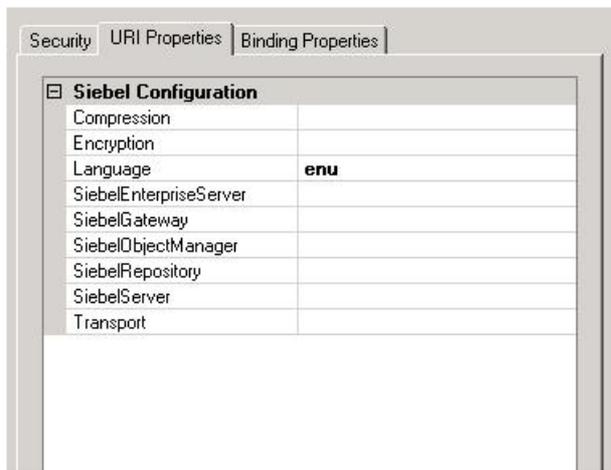
Note that WCF-based Siebel adapter does not require you to configure a physical BizTalk port for the design-time experience.

2. **Connect to the Siebel system.** In the Consume Adapter Service BizTalk Project Add-in, select the binding for Siebel (siebelBinding) and click **Configure** to start the Configure Adapter dialog box. You must specify the different connection properties in this dialog box to connect to the Siebel system.



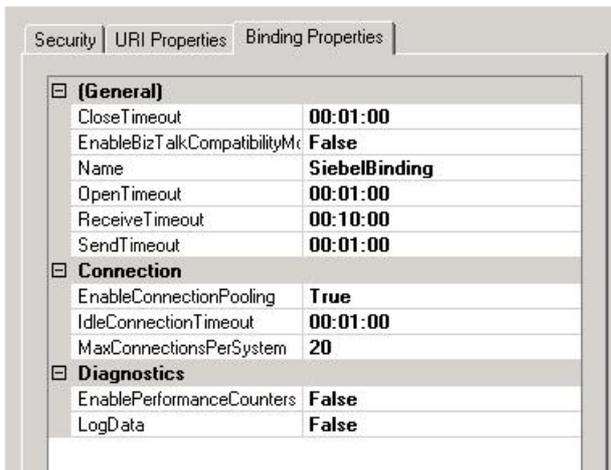
The key connection properties to connect to a Siebel system using the WCF-based Siebel adapter are:

- **Siebel Object Manager.** The name of the Siebel Object Manager on the enterprise server.
- **Siebel Enterprise Server.** The name of the Siebel Enterprise Server.
- **Siebel server.** The Siebel server. Required for all Siebel 7.5 server connections (7.5.2, 7.5.3, etc.); otherwise, do not set this parameter.
- **User name.** The user name on the Siebel system; this value is case-sensitive.
- **Password.** The password for the user on the Siebel system; this value is case-sensitive.
- **Language.** The language of the object manager.



For more information about the connection properties, see [The Siebel System Connection URI](#).

3. **Specify binding properties.** The WCF-based Siebel adapter is essentially a custom WCF binding that includes different binding properties. These binding properties govern the behavior of the adapter. These properties can be set both at design time (when generating the metadata) and at run time (when sending messages to the Siebel system).



For more information about the binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

Key Differences in Design-time Experience

The following table lists the key differences in the design-time experience for the two versions of the adapter.

Previous version of the Siebel adapter	WCF-based Siebel adapter
Add the adapter to the BizTalk Server Administration console under Platform Settings > Adapters .	The WCF-based Siebel adapter is a WCF custom binding and is configured as specific binding type of the WCF-Custom adapter. The BizTalk Server Administration console already contains the WCF-Custom adapter under Platform Settings > Adapters .
Create a physical send port to connect to the Siebel system at design time.	There is no need to configure a send port.
Use the Add Adapter Metadata option to connect to the Siebel system and retrieve metadata.	Use the Consume Adapter Service option to connect to the Siebel system and retrieve metadata.
There is no option to configure the adapter behavior during design time (using binding properties) to control metadata generation.	Set binding properties to configure the adapter behavior during design time to control metadata generation.

See Also

Concepts

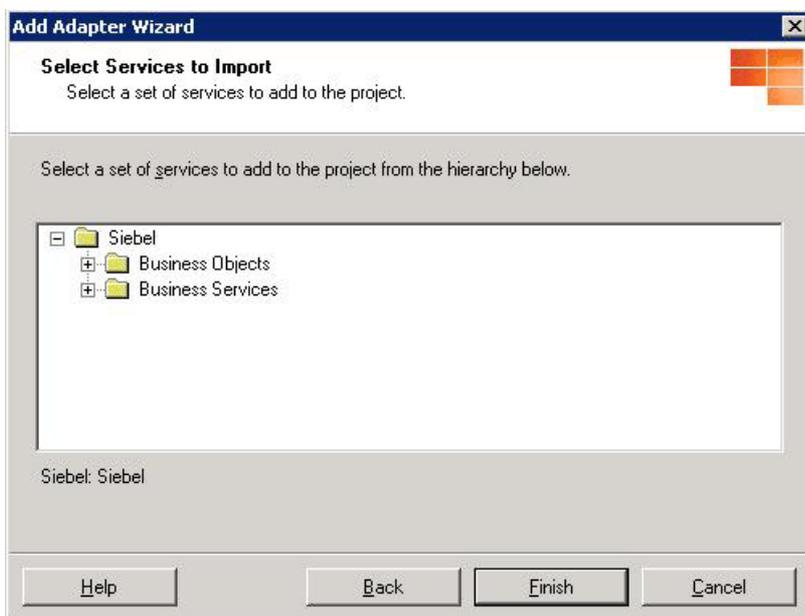
[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Differences in Metadata Retrieval Experience

Once you have connected to the Siebel system from the BizTalk project in Visual Studio, you can generate metadata to perform operations on Siebel business components or invoke Siebel business service methods. The steps involved in generating the metadata in the previous version of the Siebel adapter differ from those in the WCF-based Microsoft BizTalk Adapter for Siebel eBusiness Applications. This section discusses the differences.

Metadata Retrieval Experience for the Previous Version of the Adapter

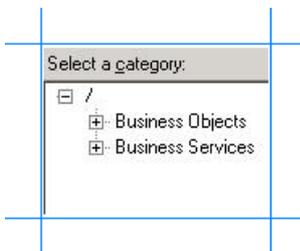
- Metadata categories.** The previous version of the adapter surfaces two categories: **Business Objects** and **Business Services**. All the business objects and their constituent business components are surfaced under **Business Objects** node. Similarly, all the business services are surfaced under the **Business Services** node. However, no operations (Insert, Query, Delete) are surfaced for the business components. Similarly, the Business Services node does not surface the different business service methods for a particular business service.



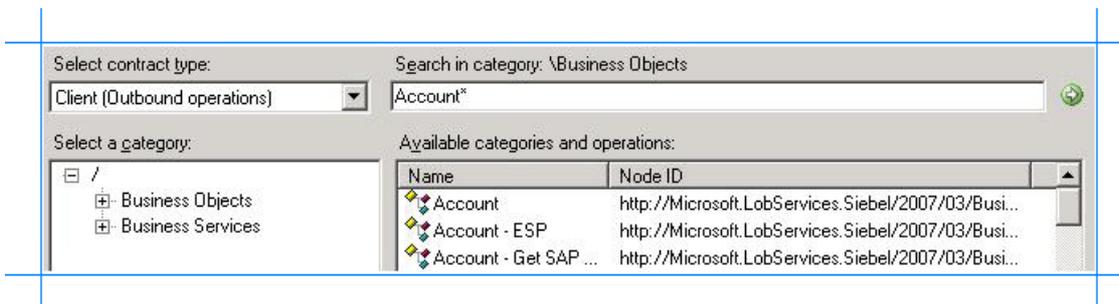
- Searching for Siebel artifacts.** The previous version of the adapter does not support searching for Siebel artifacts. You need to browse through the entire tree exposed by the adapter to find the artifact you are looking for.

Metadata Retrieval Experience for the WCF-based Adapter

- Contract type.** The WCF-based Siebel adapter only supports the contract type **Client (Outbound Operations)**. So, adapter clients can only perform outbound operations on the Siebel system. That is, adapter clients can invoke operations in the Siebel system. From a BizTalk user perspective, the contract type basically indicates the direction of message flow between BizTalk Server and the adapter. The Siebel adapter does not support inbound operations via a BizTalk Receive port.
- Metadata Categories.** The WCF-based Siebel adapter surfaces two categories: **Business Objects** and **Business Services**. All the business objects and their constituent business components are surfaced under **Business Objects** node. All operations supported on the business components are also surfaced. All the business services are surfaced under the Business Services node. The methods for a particular business service are also surfaced as operations.



- Searching for Siebel artifacts.** With the WCF-based Siebel adapter you can specify a search expression after selecting a relevant category node. For example, to search for a business object, click the Business Objects node and specify a search expression (with the same syntax as the Siebel query search expression). Note that the search expressions are case-sensitive. For more information about searching Siebel metadata using the WCF-based Siebel adapter, see [How Does the Adapter Surface Siebel Metadata?](#)



- BizTalk port binding file.** Generating schema for the WCF-based Siebel adapter using the Consume Adapter Service BizTalk Project Add-in also creates a BizTalk port binding file. You can import this file into a BizTalk application to configure a WCF-Custom port. The file essentially contains the physical port configuration including the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. This makes it significantly easier to create physical WCF-Custom ports for the project. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Key Differences in Metadata Retrieval Experience

The following table lists the key differences in the metadata retrieval experience for the two versions of the adapter.

Previous version of the Siebel adapter	WCF-based Siebel adapter
No operations are surfaced under the Business Objects and Business Services node.	The Business Objects node surfaces the business objects, business components, and the operations (Insert, Query, Delete) that can be performed on the business components. Business Services node surfaces the business nodes along with the business service methods.
Search is not supported.	Search is supported for Siebel artifacts using search expression with the same syntax as the Siebel query search expression.
No differentiation between inbound and outbound operations	Outbound operations are surfaced under the Client (Outbound Operations) category. Inbound operations are not supported

See Also
Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Differences in the Schema File Name and Namespace

The previous version of the adapter generates metadata for Siebel artifacts differently than the current version. This section discusses these differences including:

- File naming convention.
- Target namespace.

Schema File Name and Namespace for the Previous Version of the Adapter

- **File naming convention.** The previous version of the adapter generates schema files with names containing the name of the Siebel artifact. For example, generating the metadata for the Account business component creates a file with the name AccountService_Account_x5d.xsd.
- **Target namespace of the schema.** The following table lists the target namespace of the schemas generated by the previous version of the adapter.

Siebel artifact	Target namespace
Business Component	http://schemas.microsoft.com/[Siebel://Business Objects/{Business_Object}/{Business_Component}]
Business Service	http://schemas.microsoft.com/[Siebel://Business Services/{Business_Service}]

In the preceding table, the text in parentheses { } are placeholders to present the convention for the namespace. The actual namespace contains the appropriate values.

Schema File Name and Namespace for the WCF-based Adapter

- **File naming convention.** By default, the WCF-based Siebel adapter generates schema files with names containing the name of the binding, for example, SiebelBindingSchema1.xsd. However, the Consume Adapter Service BizTalk Project Add-in provides the option of specifying a custom file name while generating the schema. For more information about how to retrieve the schema, see [Retrieving Metadata for Siebel Operations in Visual Studio](#).
- **Target namespace of the schema.** The following table lists the target namespace of the schemas generated by the WCF-based Siebel adapter.

Siebel artifact	Target namespace
Business Component	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/{Business_Object}/{Business_Component}/Operation
Business Service	http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/{Business_Service}

In the preceding table, the text in parentheses { } are placeholders to present the convention for the namespace. The actual namespace contains the appropriate values.

See Also

Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Differences in Data Type Mapping

The two versions of the adapter differ in how the Siebel data type maps to the XSD date type. This section lists the data type mapping for Siebel artifacts in both versions of the adapter.

Data Type Mapping for Siebel Artifacts

The following table lists the mapping between the Siebel data types and XSD data types for Siebel artifacts for both the adapter versions. For more information about data type mapping for WCF-based Siebel adapter, see [Basic Siebel Data Types](#).

Siebel data type	XSD mapping for previous version of the adapter	XSD mapping for WCF-based adapter (for Beta 2 release)	XSD mapping for WCF-based adapter (for Nov CTP release)
DTYPE_BOOL	xsd:boolean	xsd:boolean	xsd:boolean
DTYPE_CURRENCY	xsd:string with pattern	xsd:decimal	xsd:decimal
DTYPE_DATE	xsd:date	xsd:dateTime	xsd:dateTime with pattern or xsd:string
DTYPE_DATE TIME	xsd:dateTime	xsd:dateTime or xsd:string	xsd:dateTime or xsd:string
DTYPE_ID	xsd:string	xsd:string	xsd:string
DTYPE_INTEGER	xsd:int	xsd:int	xsd:int
DTYPE_NOTE	xsd:string	xsd:string	xsd:string
DTYPE_NUMBER	xsd:int	xsd:decimal	xsd:decimal
DTYPE_PHONE	xsd:string	xsd:string	xsd:string
DTYPE_TEXT	xsd:string	xsd:string	xsd:string
DTYPE_TIME*	xsd:time	xsd:string	xsd:dateTime with pattern or xsd:string
DTYPE_UTCDATETIME	xsd:dateTime	xsd:dateTime	xsd:dateTime or xsd:string

* The supported date pattern for the November CTP release was `(0001-01-01)T(\d\d:\d\d:\d\d)(.*)`. For the Release Candidate (RC) and RTM release, the supported date pattern is changed to `(1753-01-01)T(\d\d:\d\d:\d\d)(.*)`. All other data type mappings remain unchanged.

See Also

Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Differences in Operations Surfaced for the Siebel Artifacts

The two versions of the adapter differ in the operations they expose for the Siebel artifacts. This section lists the differences in:

- Operations on business components.
- Operations on business services.

For more information about the operations supported by the WCF-based Siebel adapter, see [What Operations Can be Performed Using the Adapter?](#)

Operations on Business Components

The following table lists the operations surfaced for business components by both versions of the adapter.

Operation for previous version of the adapter	Description	Operation for WCF-based adapter	Description
Insert	Inserts one or more records into the business component.	Insert	Inserts one or more records into the business component.
InsertEx	Inserts one or more records into the business component with ViewMode support and business component preloading.		
Query	Queries records from a business component based on search and sort specifications.	Query	Queries records from a business component based on search and sort specifications, query fields, and ViewMode.
QueryWithViewMode	Queries records from a business component based on search and sort specifications with support for ViewMode.		
QueryEx	Same as QueryWithViewMode and includes business component preloading.		
QueryEx2	Same as QueryEx.		
Update	Updates one or more records in the business component.	Update	Updates one or more records in the business component with support for ViewMode.
UpdateEx	Updates one or more records in the business component with ViewMode support and business component preloading.		
Delete	Delete one or more records in the business component	Delete	Delete one or more records in the business component based on record ID or search expression and with support for ViewMode
DeleteEx	Delete one or more records in the business component with ViewMode support and business component preloading		

Operations on Business Services

- The previous version of the adapter does not surface any operations for the business service.

- The WCF-based Siebel adapter surfaces business service methods as operations that adapter clients can invoke.

See Also

Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Differences in Physical Port Configuration in the BizTalk Server Administration Console

To use the previous version of the Siebel adapter or the WCF-based Siebel adapter with BizTalk Server, you need to configure physical ports in the BizTalk Server Administration console. This section lists the differences in configuring a physical port for both versions of the adapter.

Configuring a Physical Port for the Previous Version of the Siebel Adapter

You can configure a send port to send messages to the Siebel using the previous version of the adapter. You must provide values for the configuration properties available for the send port.

Note

You cannot configure a receive port for the previous version of the Siebel adapter.

Configuring a Physical Port for the WCF-based Siebel Adapter

For the WCF-based Siebel adapter, you do not necessarily have to create a port manually. You have the option of configuring a physical port automatically by importing a port binding file that is generated while you generate metadata for any operation using the Consume Adapter Service BizTalk Project Add-in. The file essentially contains the physical port configuration including the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Several other considerations that make a physical port configuration for WCF-based Siebel adapter different from the previous version of the Siebel adapter are:

- The port type for WCF-based Siebel adapter is "WCF-Custom." So, with respect to BizTalk Server, this port uses custom WCF configurations and, primarily, a custom WCF binding. Because the WCF-based Siebel adapter is essentially a custom WCF binding, it can be used in BizTalk Server via the WCF-Custom port.
- A BizTalk WCF-Custom send port requires an "Action" to be specified in the port. This is the WCF action that is sent on the message by BizTalk before it gets to the binding. The WCF action conveys to the adapter:
 - The operation the adapter client wants to perform.
 - The Siebel artifact on which the user wants to perform the operation.

Adapter clients can specify this action in various ways. For more information on specifying an action, see [Specifying SOAP Action](#).

- BizTalk WCF-Custom send ports include binding information that, in turn, includes a binding type and binding properties for that binding type. This is a key configuration that ties the WCF-Custom port to a WCF-based Siebel adapter. For the WCF-based Siebel adapter, binding type is **siebelBinding**.
- The binding configuration properties that govern how an adapter behaves while generating metadata or while sending messages to the Siebel system. The adapter users must be aware of which binding property to set for a specific operation performed by the adapter. For more information about the binding properties, see [Working with BizTalk Adapter for Siebel Binding Properties](#).
- The configurations under the **Messages** tab in the **WCF-Custom Transport Properties** dialog box enable adapter users to alter the message that is sent or received using the WCF-Custom port. While sending messages (from BizTalk to the adapter), users can choose to send the message body or use a template to transform the XML message to a different message structure before sending it to the adapter (the WCF custom binding). While receiving messages (from the adapter to BizTalk), users can choose to get the entire SOAP envelope, the body contents of the SOAP message, or a specific node value in the incoming message referenced by an XPath query expression.

See Also

Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Migration Tutorials and Sample

The BizTalk Adapter Pack includes migration tutorials that demonstrate how to migrate BizTalk projects created using the previous version of the Siebel adapter to use the WCF-based Siebel adapter. For more information, see [Tutorial 2: Migrating BizTalk Projects](#).

The BizTalk Adapter Pack also includes a sample that demonstrates how to modify a BizTalk project created using the previous version of the Siebel adapter to send and receive messages using the WCF-based Siebel adapter. For more information, see [Samples](#).

See Also

Concepts

[Migrating BizTalk Projects Created Using the Previous Version of the Siebel Adapter](#)

Troubleshooting

The BizTalk Adapter Pack uses or depends on several different Microsoft technologies including but not limited to Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, Microsoft Visual Studio, and the Microsoft .NET Framework 3.5 SP1/.NET Framework 4. The BizTalk Adapter Pack is built on top of the WCF LOB Adapter SDK, which in turn requires .NET Framework 3.5 SP1 and .NET Framework 4. The adapters can be consumed either by writing applications using Visual Studio or by creating BizTalk applications. For issues related to each of these technologies and products, refer to the respective documentation.

This section provides information about troubleshooting the BizTalk Adapter Pack, including:

- Enabling tracing to diagnose issues with the adapters.
- Handling installation and operational issues that you might encounter when working with the adapters, including probable cause, and a resolution.
- Using performance counters to gauge adapter performance.
- Handling exceptions and errors, including probable cause, and a resolution.

In This Section

- [Diagnostic Tracing and Message Logging](#)
- [Troubleshooting Installation Issues](#)
- [Troubleshooting Operational Issues](#)
- [Troubleshooting Performance Issues](#)
- [Troubleshooting Issues with the Data Provider for Siebel](#)
- [Using Performance Counters](#)
- [Exceptions and Error Handling](#)

Diagnostic Tracing and Message Logging

Adapter clients can enable diagnostic tracing to effectively diagnose problems encountered while using the adapters. Adapter clients can activate tracing at three different levels:

- Between the adapter client and the adapter
- Within the adapter
- Between the adapter and the line-of-business (LOB) application

This section provides information about activating tracing at these levels.

Tracing Between the Adapter Client and the Adapter

Adapter clients can enable WCF tracing to trace issues between the adapter client and the adapter. WCF tracing is used to trace the input XMLs coming from the adapter client using the WCF service model and is useful in diagnosing serialization issues. WCF tracing is not used for the WCF channel model or for output messages from the adapter to the adapter client. You can activate WCF tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in `<installation drive>\Program Files\Microsoft Visual Studio <version>\Common7\IDE`.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically `BTSNTSvc.exe.config`. For BizTalk Server 2010, this file is available typically under `<installation drive>\Program Files\Microsoft BizTalk Server 2010`.
 - For a WCF service model .NET application, you must add the excerpt to the `app.config` file of your project.

To enable WCF tracing, you must add the following excerpt within the `<configuration>` tag.

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.ServiceModel.MessageLogging"
      switchValue="Verbose, ActivityTracing">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.Runtime.Serialization" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\WCFTrace.svclog" />
  </sharedListeners>
```

```

<trace autoflush="true" />
</system.diagnostics>
<system.serviceModel>
  <diagnostics>
    <messageLogging
      logEntireMessage="true"
      logMalformedMessages="false"
      logMessagesAtServiceLevel="true"
      logMessagesAtTransportLevel="false"/>
    </diagnostics>
  </system.serviceModel>

```

This saves the WCF traces to C:\log\WCFTrace.svclog. For more information about WCF tracing, see <http://go.microsoft.com/fwlink/?LinkId=91242>.

◆ Important

Make sure you mitigate potential security threats of exposing sensitive business data by enabling tracing. For recommendations see [Best Practices](#).

Tracing Within the Adapter

The adapters in the BizTalk Adapter Pack log different categories of useful information to the trace file such as errors, warnings, and information. Such information is useful in understanding the process flow within the adapter and diagnosing issues with the adapter. You can activate WCF LOB Adapter SDK and adapter tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design-time and run-time.

- **Tracing at design-time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in *<installation drive>*:\Program Files\Microsoft Visual Studio *<version>*\Common7\IDE.
- **Tracing at run-time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server applications, you must add the excerpt to the BizTalk configuration file, typically BTSNTSvc.exe.config. For BizTalk Server 2010, this file is available typically under *<installation drive>*:\Program Files\Microsoft BizTalk Server 2010.
 - For a WCF service model .NET application, you must add the excerpt to the app.config file of your project.

To enable WCF LOB Adapter SDK and adapter tracing, you must add the following excerpt within the `<configuration>` tag:

```

<system.diagnostics>
  <sources>
    <source name="Microsoft.ServiceModel.Channels" switchValue="Error">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="Microsoft.Adapters.Siebel" switchValue="Information">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\AdapterTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>

```

This would save the WCF traces to C:\log\AdapterTrace.svclog.

Tracing Between the Adapter and the LOB Application

You must enable tracing for communication between the adapter and the LOB application to diagnose issues you suspect within the LOB application. Adapters also depend on LOB tracing (client/server side) to get access to this information. The specifics of turning on LOB tracing are excluded from this document.

Additionally, the Siebel adapter provides a binding property (**LogData**), which if set to **True** and if the trace level is set to **Verbose**, the Siebel adapter logs the information flow between the adapter and the Siebel system. This information is logged along with the adapter traces in the same trace file.

For more information about this binding property, see [Working with BizTalk Adapter for Siebel Binding Properties](#).

Viewing the Traces

You can use the Windows Communication Foundation (WCF) Service Trace Viewer tool to view the traces. For more information about the tool, see <http://go.microsoft.com/fwlink/?LinkId=91243>.

Configuring Tracking for BizTalk Applications

The BizTalk Administration Console enables you to configure various tracking options for things such as send ports, receive ports. The tracking configuration settings enable you to track inbound/outbound event data, message properties, message bodies, and orchestrations. For more information about configuring tracking for BizTalk applications, see "Configuring Tracking Using the BizTalk Server Administration Console" at <http://go.microsoft.com/fwlink/?LinkId=106851>.

You can also use Health and Activity Tracking (HAT) to view historical tracked data. For more information, see "Viewing Historical or Tracked Data Using HAT" at <http://go.microsoft.com/fwlink/?LinkId=106852>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Installation Issues

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter Pack installation copies the product binaries on the computer and registers the bindings for each adapter. This section discusses troubleshooting techniques to resolve installation errors.

Setup Logging

The BizTalk Adapter Pack setup program performs the standard task of installing the BizTalk Adapter Pack. Additionally, the setup also performs certain custom actions such as registering the adapter bindings. You can log messages for both the standard as well as custom actions performed by the setup.

- The BizTalk Adapter Pack setup installs the adapter specific files using an MSI. Hence, the logging for the setup will be the standard MSI logging.
- Logs for the custom actions performed by the setup program are available at %TEMP%\adaptersetup.log. If the tracing to the log file fails, the traces are also available in the event log.

Known Issues

Following are the most common errors you may encounter while installing the BizTalk Adapter Pack, their probable cause, and resolution.

- [Setup fails to register adapter bindings](#)

Setup fails to register adapter bindings

Problem

The Microsoft BizTalk Adapter Pack setup wizard fails to register the Siebel adapter binding or the Data Provider for Siebel, but proceeds with the adapter installation.

Cause

This might result due to problems with WCF installation, Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK installation, or the machine.config being corrupt. The adapter bindings are written to the machine.config file.

Resolution

You should manually register the Siebel adapter binding and Data Provider for Siebel.

To register the adapter binding or the data provider

1. Navigate to the machine.config file on the computer. For example, on a 32-bit platform, the machine.config is available under <system drive>:\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG.

In this path, <version> is the version of the .NET Framework. For example, for Microsoft .NET Framework 3.5 SP1, the version will be v2.0.50727.

2. Open the file using a text editor.
3. To register the Siebel adapter binding:
 - a. Search for the element "system.serviceModel" and add the following under it:

```
<client>
  <endpoint binding="siebelBinding" contract="IMetadataExchange" name="siebel" />
</client>
```

- b. Search for the element "bindingElementExtensions" under system.serviceModel\extensions.

- c. Look for the missing Siebel adapter binding. Add the following section under the "bindingElementExtensions" node.

For Siebel adapter, add:

```
<add name="siebelAdapter" type="Microsoft.Adapters.Siebel.SiebelAdapterExtensionElement,Microsoft.Adapters.Siebel, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

- d. Search for the element "bindingExtensions" under system.serviceModel\extensions.

- e. Look for the missing Siebel adapter binding. Add the following sections under the "bindingExtensions" node.

For Siebel adapter, add:

```
<add name="siebelBinding" type="Microsoft.Adapters.Siebel.SiebelAdapterBindingSection,Microsoft.Adapters.Siebel, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

Note

For information about how to determine the public key, see [Determining the Public Key and Version](#).

- 4. To register the Data Provider for Siebel:

- a. Search for the element DbProviderFactories under the system.data node.
- b. Look for the missing Data Provider for Siebel. Add the following section under the DbProviderFactories node.

For Data Provider for Siebel, add:

```
<add name="SiebelClient Data Provider" invariant="Microsoft.Data.SiebelClient" description=".NET Framework Data Provider for Siebel eBusiness Applications" type="Microsoft.Data.SiebelClient.SiebelProviderFactory,Microsoft.Data.SiebelClient, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

- 5. Save and close the machine.config file.

Determining the Public Key and Version

Perform the following steps to determine the public key for Siebel adapter or Data Provider for Siebel.

To determine the public key

1. Navigate to the Windows directory, typically C:\WINDOWS\assembly.
2. Right-click the DLL for which you want the public key and select **Properties**. The following table lists the name of the DLLs for each adapter and provider.

Adapter/ADO Provider	Name of the DLL
Siebel adapter	Microsoft.Adapters.Siebel
Data Provider for Siebel	Microsoft.Data.SiebelClient

3. On the **General** tab, the value against the **Public Key Token** label specifies the public key for the DLL. Similarly, value against the **Version** label specifies the version number for the DLL.
4. Copy the public key, and then click **Cancel**.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Operational Issues

This topic was last updated on: June 05, 2009

This section provides a centralized location for information about operational issues you might encounter when using the Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Enabling Tracing

For information about tracing support in the Siebel adapter, see [Diagnostic Tracing and Message Logging](#).

Known Issues

The following are some issues and recommended solutions that you might encounter while using the Siebel adapter.

- [Error in loading the adapter bindings](#)
- [The Siebel adapter does not display in the list of adapters in BizTalk Server Administration console](#)
- [Error while connecting to the Siebel system](#)
- [Error while retrieving XMLs with more than 65536 nodes](#)
- [Error while specifying a connection URI for a WCF-Custom port in BizTalk](#)
- [Error while performing operation on the Siebel system](#)
- [XmlReaderParsingException due to an incorrect operation name in the specified action](#)
- [Application using the Siebel adapter does not terminate](#)
- [Siebel adapter may hang if the Siebel server is restarted](#)
- [The adapter does not recognize the action on the physical port even though you use the binding file generated by the Consume Adapter Service add-in to create the ports](#)
- [Siebel adapter does not handle Siebel objects with XML encoded strings in the name](#)
- [Error with RootNode TypeName in BizTalk Projects](#)
- [Invalid binding warning when using the adapter in Visual Studio 2010](#)

Error in loading the adapter bindings

Problem

When you try to start the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in, the GUI gives the following error:

```
There was an error loading the binding, <binding name>, from your system configuration.ConfigurationErrorsException: Exception has been thrown by the target of an invocation.
```

Cause

When you start the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, WCF loads the adapter bindings for all the installed adapters. In turn, the adapter bindings are dependent on the specific enterprise application client software. So, you could face this issue for one or both of the following reasons:

- The required LOB client software is not installed on the computer where you installed the adapter.
- You did a "Typical" or "Complete" installation of the adapter, which installs all the adapters in the BizTalk Adapter Pack. However, the client libraries might be installed for only one enterprise application. As a result, the GUI fails to load the bindings for the other adapters.

Resolution

- Make sure the required client versions are installed on the computer where you installed the BizTalk Adapter Pack.
- Make sure you do a custom installation of the adapters to install only the adapter you need.

The Siebel adapter does not display in the list of adapters in BizTalk Server Administration console

Problem

Unlike the earlier version of the adapters that shipped with BizTalk Server, the Siebel adapter that shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console.

Cause

The latest Siebel adapter is a WCF custom binding. So, although the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF custom bindings and hence, does not display the WCF-based Siebel adapter.

Resolution

You can explicitly add the Siebel adapter to the BizTalk Server Administration console by following the steps mentioned in [Adding the Siebel Adapter to BizTalk Server Administration Console](#).

Error while connecting to the Siebel system

Problem

The Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in gives the following error when you try to connect to the Siebel system:

```
Connecting to the system LOB has failed. Retrieving the COM class factory for component with CLSID {ID} failed due to the following error: 80040154
```

Cause

The Siebel Web client might not be installed on the computer.

Resolution

Make sure the supported version of the Siebel Web client is installed on the computer. Refer to the installation guide for supported client and server versions for Siebel. The installation guide is available at <system drive>\Program Files\Microsoft BizTalk Adapter Pack\Documents.

Error while retrieving XMLs with more than 65536 nodes

Problem

The adapter gives the following error while retrieving XML output that has more than 65536 nodes.

```
Maximum number of items that can be serialized or deserialized in an object graph is '65536'.Change the object graph or increase the MaxItemsInObjectGraph quota.
```

Cause

The adapter cannot serialize and deserialize an object with more than 65536 items.

Resolution

You can fix this issue by setting the `maxItemsInObjectGraph` parameter. You can set this in any of the following two ways:

- Set this parameter by changing the `maxItemsInObjectGraph` parameter in the `ServiceBehavior` attribute on your service class.
- Add the following to your application's `app.config` file.

```
<behaviors>
  <endpointBehaviors>
    <behavior name="NewBehavior">
      <dataContractSerializer maxItemsInObjectGraph="65536000" />
    </behavior>
  </endpointBehaviors>
</behaviors>
```

A sample `app.config` will look like:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.serviceModel>
    <behaviors>
      <endpointBehaviors>
        <behavior name="NewBehavior">
          <dataContractSerializer maxItemsInObjectGraph="65536000" />
        </behavior>
      </endpointBehaviors>
    </behaviors>
    <client>
      <endpoint behaviorConfiguration="NewBehavior" binding="siebelBinding"
        contract="IOutboundContract" name="siebel_ICalculator" />
    </client>
  </system.serviceModel>
</configuration>
```

Error while specifying a connection URI for a WCF-Custom port in BizTalk

Problem

BizTalk Server gives the following error when you specify a connection URI to connect to the Siebel system.

```
Error saving properties.
(System.ArgumentException) The specified address is invalid.
(System.ArgumentException) Invalid address;
"<connection URI>" is not a well-formed absolute uri.
```

Cause

The connection URI does not adhere to the standard encoding format. For example, the value for a parameter might contain a space.

Resolution

Make sure the connection URI you specify adheres to the standard encoding format. For example, a blank space must be replaced by "%20".

Error while performing operation on the Siebel system

Problem

The adapter gives the following error when performing any operation on the Siebel system using BizTalk Server.

- For BizTalk Server 2010

```
System.ArgumentNullException: Value cannot be null.
```

Cause

The WCF action for the message is not specified. WCF requires a SOAP action to be specified for every operation, which informs the adapter about the operation to be performed on the LOB application.

Resolution

Specify the SOAP action in the send port or as a message context property in a BizTalk orchestration. For instructions, see [Specifying SOAP Action](#). See [Technical Reference](#) to see a list of actions for each operation.

XmlReaderParsingException due to an incorrect operation name in the specified action

Problem

The BizTalk Server Administration console gives the following error when sending messages to a Siebel system:

```
Microsoft.ServiceModel.Channels.Common.XmlReaderParsingException: Invalid argument:
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="operation_name" Action="<action>" />
</BtsActionMapping>
```

Cause

If you configure a WCF-Custom port by importing the port binding file created by the Consume Adapter Service BizTalk Project Add-in, the action in the port is specified in the following format:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="http://MyService/Svc/Op1" />
</BtsActionMapping>
```

In the preceding format, the operation name is governed by the operation you chose while generating the schema. For example, if you generated schema for a Query operation on a Siebel business component, the operation name in the action will be "Query". However, the operation name in the logical port created in the BizTalk orchestration in Visual Studio might be different.

Resolution

Make sure the operation names in both the logical port (in the BizTalk orchestration in Visual Studio) and the physical port (in BizTalk Server Administration console) are the same.

Application using the Siebel adapter does not terminate

Problem

An application that uses the Siebel adapter with Siebel client version 7.5 does not terminate.

Cause

This is because of a Siebel client issue where the process does not terminate when logging off from a Siebel server.

Resolution

Make sure you have the patch 7.5.3.17 installed for the Siebel server, along with the quick fix QF0H05.

Siebel adapter may hang if the Siebel server is restarted

Problem

If the Siebel server is restarted while the Siebel adapter is sending a message to the Siebel server using, the Siebel adapter may hang.

Resolution

Restart the BizTalk application host instance. To do so from the BizTalk Server Administration console, in the console tree expand **BizTalk Group**, expand **Platform Settings**, and then click **Host Instances**. From the right pane, right-click the host name, and then select **Restart**.

The adapter does not recognize the action on the physical port even though you use the binding file generated by the Consume Adapter Service add-in to create the ports

Problem

After you use the Consume Adapter Service Add-in to generate schema for a specific operation on the Siebel system, the add-in also creates a port binding file. You can import this binding file using the BizTalk Server Administration console to create physical ports in BizTalk Server. However, when you send messages to the Siebel system using such ports, the adapter fails to understand the action specified on the port and gives an error similar to the following:

```
Microsoft.ServiceModel.Channels.Common.UnsupportedOperationException: Incorrect Action
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="op_name" Action="<action>" />
</BtsActionMapping>. Correct the specified Action, or refer to the documentation on the allowed formats for the Actions.
```

Cause

When you create logical ports in a BizTalk orchestration, you specify certain names for the operations on those ports or you just use the default names like Operation_1, Operation_2, etc. However, in the binding file generated by the Consume Adapter Service Add-in, the operation name is same as the name of the operation for which you generate metadata. For example, if you generate metadata for Insert operation on the Account business component, the action will be set to the following:

```
<Operation Name="Insert" Action="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert" />
```

When you import the binding file, the same action is set on physical port. So, the operation names on the logical port (Operation_1, Operation_2, etc) do not match the operation names specified in the action on the physical port, resulting in an error.

Resolution

Make sure the operation name in the logical port is the same as the operation name specified as part of the action in the physical port. Do one of the following:

- Change the operation name in the logical port in BizTalk orchestration from Operation_1, etc. to the operation for which you generate metadata, for example Insert.
- Change the operation name in the action on the physical port to the operation name in the logical port. For example, you could change the action in the physical port to resemble the following:

```
<Operation Name="Operation_1" Action="http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert" />
```

Siebel adapter does not handle Siebel objects with XML encoded strings in the name

Problem

The Siebel adapter cannot perform operations involving Siebel objects (business objects, business components, business services, picklist, methods, fields, arguments, etc) that have XML encoded strings in their name. For example, the Siebel adapter will not be able to invoke a business service method with the name Time_x0020_Stamp.

Resolution

Make sure the Siebel objects do not contain XML encoded strings in their name.

Error with RootNode TypeName in BizTalk Projects

Problem

In a BizTalk project in Visual Studio, if the schemas generated from the Consume Adapter Service Add-in contains invalid characters or reserved words for the **RootNode TypeName** property, the following error will occur while compiling the project:

```
Node <node reference> - Specify a valid .NET type name for this root node.  
The current .NET type name of this root node is invalid (it is a reserved BizTalk Keyword or is an invalid C# identifier).
```

Resolution

1. Right-click the rood node referenced in the error and select **Properties**.
2. For the **RootNode TypeName** property, remove any illegal characters or reserved words, for example, dot (.

Invalid binding warning when using the adapter in Visual Studio 2010

Problem

When you use the adapter to create an application in Visual Studio 2010 and you open the configuration file (app.config) generated by the adapter, you see a warning similar to the following:

```
The element 'bindings' has invalid child element 'siebelBinding'. List of possible elements expected: 'basicHttpBinding, custom Binding, ...
```

Cause

This warning appears because the Siebel adapter binding, `siebelBinding`, is not a standard binding shipped with the Windows Communication Foundation (WCF).

Resolution

You can safely ignore this warning.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Performance Issues

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve performance issues that you might encounter when using Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Known Issues

The following are the most common performance issues you might encounter when using the Siebel adapter, along with their probable cause and resolution.

- [Slowdown or stall in throughput when using the adapter with BizTalk Server](#)

Slowdown or stall in throughput when using the adapter with BizTalk Server

Problem

When using the Siebel adapter with BizTalk Server, the number of messages sent or received by the adapter slows down or comes to a stall.

Cause

The **EnableBizTalkCompatibilityMode** binding property is not set on the WCF-Custom send or receive port in BizTalk Server Administration console.

Resolution

Set the **EnableBizTalkCompatibilityMode** binding property to True. For more information about this property, see [Working with BizTalk Adapter for Siebel Binding Properties](#). For instructions on how to set a binding property, see [Specifying Binding Properties](#).

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Issues with the Data Provider for Siebel

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve errors that you might encounter when using the .NET Framework Data Provider for Siebel eBusiness Applications (Data Provider for Siebel).

Known Issues

The following are the most common errors you might encounter when using Data Provider for Siebel, along with their probable cause and resolution.

- [Data Provider for Siebel may give "component 'DataReader Source' \(380\)" error](#)

Data Provider for Siebel may give "component 'DataReader Source' (380)" error

Problem

While performing a SELECT query on a Siebel business component, the Data Provider for Siebel may give a "component 'DataReader Source' (380)" error.

Cause

The Data Provider for Siebel gives this error if the value received from the Siebel system for a parameter exceeds the maxLength property for the parameter.

See Also

Other Resources

[Troubleshooting](#)

Using Performance Counters

BizTalk Adapter Pack clients can use the performance counters to gauge the performance of the adapters. The BizTalk Adapter Pack setup program creates the performance counter category "Microsoft BizTalk Adapter for Siebel eBusiness Applications" along with the Adapter Pack installation.

The LOB Time (Cumulative) Performance Counter

The **BizTalk .NET Adapter for Siebel** category has one performance counter called "LOB Time (Cumulative)". This performance counter denotes the time, in milliseconds, that the LOB client library takes to complete an action that the adapter initiates. The Siebel adapter creates an instance of the performance counter for each action, for a specific Siebel server name. The instances are created in the following pattern:

```
<process id>:<app domain id>:<endpoint id>:<action id>
```

In case of the Siebel adapter, the endpoint id is the name of the Siebel server, as specified in the connection URI. The action id could be any action performed by the Siebel adapter such as Login, Logoff, Metadata, <business component name>. <operation>, <business service name>.<business service method>. If the preceding naming convention results in a name that exceeds 127 characters only the action ID is displayed in the following format:

```
:::<action id>
```

If :::<action id> also exceeds 127 characters, it is trimmed down to 127 characters.

The performance counter is initialized only after the adapter makes the first call to the Siebel system. Also, the **InstanceLifetime** property of the performance counter is set to 'Process', which means that the performance counter ceases to exist as soon as the program that creates the counter terminates. For more information about the **InstanceLifetime** property, see "PerformanceCounter.....InstanceLifetime Property" at <http://go.microsoft.com/fwlink/?LinkId=104181>.

Note

The precision of the LOB Time (Cumulative) performance counter is 16 milliseconds.

Enabling Performance Counters

The performance counters can be enabled or disabled by setting the binding property **EnablePerformanceCounters**. Set **EnablePerformanceCounters** binding property to **True** to enable performance counters. To disable performance counters, set **EnablePerformanceCounters** to **False**. By default, **EnablePerformanceCounters** is set to **False**.

Performance Counters and the WCF LOB Adapter SDK

Changing the value of the **EnablePerformanceCounters** binding property changes the value of the corresponding performance counter for WCF LOB Adapter SDK. Also, the binding property for the WCF LOB Adapter SDK is static, whereas that for the BizTalk Adapter Pack is dynamic. Hence, if there are two instances of the Siebel adapter binding in the AppDomain, and the **EnablePerformanceCounters** binding property is set to **True** in one and **False** in the other, the adapter-specific performance counter will be enabled in one and disabled in the other. However, because the binding property for the WCF LOB Adapter SDK is static, it will either be set to **True** or **False**, depending on what value was specified last.

See Also

Other Resources

[Troubleshooting](#)

Exceptions and Error Handling

This section lists the exceptions thrown by the Microsoft BizTalk Adapter for Siebel eBusiness Applications. These can contain:

- An inner exception, which is a system exception that the .NET Framework throws
- An LOB exception that the LOB client library throws.

For more information about the inner exception, refer to the respective .NET Framework or Siebel documentation. The exception also contains a detailed error message that helps in resolving the problem. Note that the list of exceptions mentioned here is not comprehensive.

Exception	Possible Cause/Error Description
ConnectionException	The adapter throws this exception if it is unable to establish a connection or close an existing connection to a Siebel system.
CredentialsException	The adapter throws this exception if the adapter client does not specify a user name or password to connect to a Siebel system.
MetadataException	The adapter throws this exception if it fails to retrieve metadata for Siebel artifacts.
XmlReaderParsingException	The adapter throws this exception if the input information provided by the adapter clients to invoke an operation in the Siebel system, is either incomplete or incorrect.
XmlReaderGenerationException	The adapter throws this exception if it is unable to generate output for an operation executed in a Siebel system.
TargetSystemException	The adapter throws this exception if the Siebel COM API, which the adapter uses to interface with the Siebel system, throws an exception. The inner exception contains the exception thrown by the Siebel COM API.

See Also

Other Resources

[Understanding BizTalk Adapter for Siebel eBusiness Applications Troubleshooting](#)

Technical Reference

This section covers technical reference information about Microsoft BizTalk Adapter for Siebel eBusiness Applications.

In This Section

- [Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)
- [Metadata Node IDs](#)
- [Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Siebel eBusiness Applications](#)
- [Glossary](#)

Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications

The Microsoft BizTalk Adapter for Siebel eBusiness Applications is a Windows Communication Foundation (WCF) custom binding. It exposes operations that applications can invoke on a Siebel system. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel.

As a WCF service, the Siebel adapter exposes metadata for its operations and data types by using standard WCF mechanisms. The sections in this topic describe the XML structure of the messages and data types that the Siebel adapter uses.

In This Section

- [Basic Siebel Data Types](#)
- [Message Schemas for Business Component Operations](#)
- [Message Schemas for Business Service Operations](#)
- [Message Schema for Picklist Operations](#)
- [Message Versioning Support](#)

See Also

Other Resources

[Technical Reference](#)

Basic Siebel Data Types

This section describes how Siebel data types are supported on the Microsoft BizTalk Adapter for Siebel eBusiness Applications.

Supported Siebel Data Types

The following table shows the Siebel data types that the Siebel adapter supports and how they are represented by the adapter for BizTalk (XSD type) and in the WCF service model (.NET type). For the types marked with an asterisk, see the note following the table.

Data Type	XSD type	.NET type	Description
DTYPE_BOOLEAN	xsd:boolean	Boolean	-
DTYPE_CURRENCY	xsd:decimal	Decimal	-
DTYPE_DATE*	xsd:dateTime*	DateTime	<p>The value must not be Coordinated Universal Time (UTC).</p> <ul style="list-style-type: none"> For xsd:dateTime, values are expected to follow this pattern: "(\\d\\d\\d\\d-\\d\\d-\\d\\d)T(00:00:00)(.*)". For DateTime objects, DateTime.Kind must be DateTimeKind.Unspecified. <p>The time component will be ignored by the adapter.</p> <p>For outbound messages, the adapter performs a runtime validation to ensure that the value specified is not UTC (z or UTC offset). If that validation fails, the adapter throws an exception.</p> <p>When this type is exposed as xsd:string (based on rules explained below):</p> <ul style="list-style-type: none"> The format is determined by the underlying database. No runtime validation is performed on the value.
DTYPE_DATETIME*	xsd:dateTime*	DateTime	<p>The value can contain both date and time components and must not be UTC.</p> <ul style="list-style-type: none"> For DateTime objects, DateTime.Kind must be DateTimeKind.Unspecified. <p>For outbound messages, the adapter performs a run-time validation to ensure that these conditions are met; if the validation fails, the adapter throws an exception.</p> <p>When this type is exposed as xsd:string (based on rules explained below):</p> <ul style="list-style-type: none"> The format is determined by the underlying database. No run-time validation is performed on the value.
DTYPE_ID	xsd:string	String	-
DTYPE_INTEGER	xsd:int	Int32	-

DTYPE_NUMBER	xsd:string	String	-
DTYPE_NUMBER	xsd:decimal	Decimal	-
DTYPE_PHONE	xsd:string	String	-
DTYPE_TEXT	xsd:string	String	-
DTYPE_TIME	xsd:dateTime*	Date Time	<p>The value must not be UTC.</p> <ul style="list-style-type: none"> For xsd:dateTime, values are expected to follow this pattern: (1753-01-01)T(\d\d:\d\d:\d\d)(.*)". For DateTime objects, DateTime.Kind must be DateTimeKind.Unspecified. <p>For outbound messages, the adapter performs a runtime validation to ensure that the value specified is not UTC (z or UTC offset). If that validation fails, the adapter throws an exception.</p> <p>When this type is exposed as xsd:string (based on the rules explained below):</p> <ul style="list-style-type: none"> The format is determined by the underlying database. No run-time validation is performed on the value.
DTYPE_UTC_DATETIME	xsd:dateTime*	Date Time	<p>The value can contain both date and time components and must be UTC.</p> <ul style="list-style-type: none"> For xsd:dateTime, the value must be expressed in UTC ('Z' notation or UTC offset). For DateTime objects DateTime.Kind must be DateTimeKind.Utc. <p>For outbound messages, the adapter performs a run-time validation to ensure that these conditions are met; if the validation fails, the adapter throws an exception.</p> <p>When this type is exposed as xsd:string (based on rules explained below):</p> <ul style="list-style-type: none"> The format is determined by the underlying database. No run-time validation is performed on the value.

The following are the Business Service method argument types:

Date

The same as DTYPE_DATE.

Number

The same as DTYPE_NUMBER.

String

The same as DTYPE_TEXT.

Hierarchy

Corresponds to XSD type xsd:string, and to .Net type String. In XML messages, this has to be placed in a CDATA node.

Integration Object

The same as Hierarchy.

*The adapter determines whether to use `xsd:dateTime` or `xsd:string` to represent `DTYPE_DATE`, `DTYPE_DATETIME`, `DTYPE_TIME`, and `DTYPE_UTCDATETIME` fields in business components in the following manner.

1. If the business component field has one of the preceding data types, the adapter will expose it as the `xsd:dateTime` type (in .Net this maps to the `DateTime` type).
2. If the business component field has no data type, the adapter will expose it as `xsd:string` (in .Net this maps to the `String` type).

Supported Facets for the XML Schema Types

The Siebel adapter supports the following facets for the XML schema types.

Siebel Type	Facet
DTYPE_BOOL	None
DTYPE_CURRENCY	Precision (22), Scale
DTYPE_DATE	(\d\d\d\d-\d\d-\d\d)T(00:00:00)(.*)
DTYPE_DATETIME	None
DTYPE_ID	MaxLength (15)
DTYPE_INTEGER	Precision (22)
DTYPE_NOTE	MaxLength (16384)
DTYPE_NUMBER	Precision (22), Scale
DTYPE_PHONE	MaxLength (40)
DTYPE_TEXT	MaxLength (2048)
DTYPE_TIME	(1753-01-01)T(\d\d:\d\d:\d\d)(.*)
DTYPE_UTCDATETIME	None

The following are some rules that govern how and when the facets, and their values, are published:

If the `Length` attribute of the field is set to a value greater than zero and less than or equal to the maximum value (specified in parentheses in the preceding table):

- The `Precision` facet is published as follows:
 - If the `Precision` attribute is set for the field, the same value is published as `Precision` facet.
 - If the `Precision` attribute is not set for the field, the `Length` value is published as the `Precision` facet.
- The `Scale` facet is published only if both:
 - The `Precision` attribute has been published
 - The `Scale` attribute is set for the field to a value greater than zero and less than the value published as part of the

Precision facet

- The MaxLength facet is the value specified for the Length attribute. This is picked up from the field definition repository. In case the length is not specified in the field definition repository, the value specified in parentheses in the preceding table gets published.

Special Cases Related to Siebel Data Types

The following rules affect the business component field facets based on the context of the operation in which they are used. These rules are applicable for INSERT and UPDATE operations only. For QUERY operations, all business component fields are exposed to the user.

Business component field marked as REQUIRED in Siebel

Even if a business component field is marked as REQUIRED in the Siebel system but the pre-default or post-default values are set for the field, Siebel adapter marks the field as OPTIONAL. Hence if a user provides a value to be inserted or updated, the adapter processes that value. If no value is provided, Siebel uses the pre-default/post-default values.

Business component field NOT marked as READ ONLY in Siebel

If a business component field is NOT marked as READ ONLY, the Siebel adapter exposes it as a writable field. However, there are a couple of exceptions to this rule. These are:

- If the business component field is a **Calculated** field in Siebel, it will not appear in the Insert or Update operations because Siebel will automatically take care of **Calculated** fields.
- If the business component field is obtained through an explicit join (table join on another table), it is generally read only. However Siebel allows data to be written to this field if it is a picklist field. Hence, if the business component field is from an explicit join and the field is NOT a picklist field, then it will NOT appear in the Insert or Update operations because adapter clients cannot write data into such fields.

Data type of a field not specified in the Business Component

If the data type of a field is not specified in the Business Component, the Siebel adapter exposes the field metadata using the following heuristics.

- If the field is a special field (i.e. picklist or join), the Siebel adapter will look up the mapped field in the destination Business Component. If that field has a type associated with it, the Siebel adapter will expose that as the type of the field. However, if that type is DTYPE_DATE, DTYPE_TIME, DTYPE_DATETIME, or DTYPE_UTCDATETIME, the Siebel adapter will expose the field as the xsd:string type. If the mapped field doesn't have an associated type, the Siebel adapter will expose the original field as the xsd:string type.
- If the field is not a picklist or join field, the Siebel adapter will expose it as the xsd:string type.

Data type, field length, or precision of a parent business component is not available

If the data type, length, or field precision of a parent business component (a business component that has a child business component based on picklists or MVLs), the Siebel adapter obtains the information about the data type, length, precision, and scale from the picklist business component or the MVL business component.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)

Message Schemas for Business Component Operations

Message Schemas for Siebel Business Component Operations

The following table lists the message schemas for the operations exposed by the Siebel adapter on Siebel business components.

Operation	XML Structure	Description
Insert	<p>Insert message:</p> <pre><Insert xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <ArrayOf[BC]InsertRecord> <[BC]InsertRecord> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </[BC]InsertRecord> ... </ArrayOf[BC]InsertRecord> </Insert></pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[FIELD1_NAME] = Business component field name; for example, Account_x0020_Status.</p> <p>Insert response message:</p> <pre><InsertResponset xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <InsertResult> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID1]</string> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID2]</string> ... </InsertResult> </InsertResponse></pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03"</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[ID] = Record ID; for example, 42-6ODBL.</p>	<p>Inserts one or more records into a business component.</p> <p>The response contains the record IDs of all records inserted.</p> <p>If at least one insertion is successful, then no exception is thrown and the response will contain the IDs of the successfully inserted records. However, if all insertions fail, an exception will be thrown (XmlReaderParsingException or TargetSystemException).</p>

<p>Query message:</p>	<pre> <Query xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <ViewMode>[View_mode]</ViewMode> <[BC]QueryInputRecord> <SearchExpr>[Search_expr]</SearchExpr> <SortSpec>Sort_spec</SortSpec> <QueryFields> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[Field1_name]</string> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[Field2_name]</string> ... </QueryFields> </[BC]QueryInputRecord> </Query> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[View_mode] = View mode to apply. This is an optional parameter. See the Siebel documentation for an explanation of possible values.</p> <p>[Search_expr] = Siebel search expression; for example, [Name] LIKE "3Com*".</p> <p>[Sort_spec] = Siebel sort specification; for example, Name (DESC), Location.</p> <p>[FIELD1_NAME] = Business component field name; for example, Account Status.</p> <p>Query response message:</p> <pre> <QueryResponse xmlns=" [VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <QueryResult> <[BC]QueryRecord> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </[BC]QueryRecord > </QueryResult> </QueryResponse> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[FIELD1_NAME] = Business component field name; for example, Account_x0020_Status.</p>	<p>Queries one or more records from the target business component. The Query operation takes the following parameters:</p> <p><SearchExpr> All records under the target business component are compared against the search expression and matching records are returned.</p> <p><SortSpec> The sort specification determines the order in which records that match the search expression are returned. This parameter is optional.</p> <p><QueryFields> A list of field names that specifies the fields in the target business component that should be retrieved by the query operation. Only the fields in this list are returned in each record. Each field should be specified by using the original name of the field in the business component, not the XML-encoded name; for example, "First Name" instead of "First_x0032_Name". This parameter is optional. If query fields are not specified, then all fields are returned.</p>
-----------------------	---	---

Update message:	<pre> <Update xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <ViewMode>[View_mode]</ViewMode> <ArrayOf[BC]UpdateRecord> <[BC]UpdateRecord> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... <Id>[Record_ID]</Id> </[BC]UpdateRecord> ... </ArrayOf[BC]UpdateRecord> </Update> </pre>	<p>Update one or more records specified by a list of update records.</p> <p>Each update record contains a list of the fields to be updated and a single, mandatory <Id> element that identifies the record to be updated.</p> <p>The response contains a list of the record IDs of all of the records updated.</p> <p>If at least one update is successful, then no exception is thrown and the response will contain the IDs of the successfully updated records. However, if all updates fail, an exception will be thrown (XmlReader ParsingException or TargetSystemException).</p>
	<p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[View_mode] = View mode to apply when querying the records that need to be updated. This is an optional parameter. See the Siebel documentation for an explanation of possible values.</p> <p>[FIELD1_NAME] = Business component field name; for example, Account_x0020_Status.</p> <p>[Record_ID] = Record ID of a record to be updated; for example, 42-60DBL.</p> <p>Update response message:</p> <pre> <UpdateResponse xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <UpdateResult> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID1]</string> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID2]</string> ... </UpdateResult> </UpdateResponse> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[ID] = Record ID; for example, 42-60DBL.</p>	

Delete message:	<pre> <Delete xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <ViewMode>[View_mode]</ViewMode> <Id> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID1]</string> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID2]</string> ... </Id> <SearchExpr>Search_expr</SearchExpr> </Delete> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[View_mode] = View mode to apply when querying the records that need to be deleted. This is an optional parameter. See the Siebel documentation for an explanation of possible values.</p> <p>[ID] = Record ID; for example, 42-6ODBL.</p> <p>Search_expr = Siebel search expression; for example, [Name] LIKE "3Com*".</p> <p>Delete response message</p> <pre> <DeleteResponse xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <DeleteResult> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID1]</string> <string xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">[ID2]</string> ... </DeleteResult> </DeleteResponse> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component; for example, Account.</p> <p>[ID] = Record ID; for example, 42-6ODBL.</p>	<p>Delete one or more records matching a set of record IDs or a search expression.</p> <p>◆ Important The Delete operation takes either a search expression or a set of record IDs. You cannot specify both parameters in the same Delete operation.</p> <p>The response contains a list of the record IDs of all of the rows deleted.</p> <p>If at least one deletion is successful, then no exception is thrown and the response will contain the IDs of the successfully deleted records. However, if all deletions fail, an exception will be thrown (XmlReaderParsingException or TargetSystemException).</p>
Associate message:	<pre> <Associate xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation"> <ViewMode>[View_mode]</ViewMode> <ParentSearchExpr>Parent_search</ParentSearchExpr> <ParentMVGField>Field_name</ParentMVGField> <ChildSearchExpr>Child_search</ChildSearchExpr> </Associate> </pre> <p>The Associate operation is surfaced only for business components with multi-value group (MVG) fields. It associates records by specifying search expressions for parent and child records.</p> <p>The Associate operation:</p> <ul style="list-style-type: none"> • Associates a selected parent business component record with one selected child business component record 	

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[BO] = Name of the business object; for example, Account.

[BC] = Name of the business component; for example, Account.

[View_mode] = View mode to apply to the query for parent and child business component records. If a value is specified, that value will be applied to both the parent search and the child search. This is an optional argument. For an explanation of possible values, see the Siebel documentation.

[Par_search] = Search expression matching exactly one parent business component record; for example, [Id] LIKE AB-12345.

[Chld_search] = Search expression matching exactly one child business component record that will be associated with the parent business component record; for example, [Id] LIKE CD-12345.

[Field_name] = Multi-value group field name of the parent business component; for example, 'Bill To First Name' in the Account business component.

Associate response message:

```
<AssociateResponse xmlns=
"[VERSION]/BusinessObjects/[BO]/[BC]/Operation">
  <AssociateResult>
    <ChildID>[CHILD_ID]</
ChildID>
    <ParentID>[PARENT_ID]
</ParentID>
  </AssociateResult>
</AssociateResponse>
```

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[BO] = Name of the business object; for example, Account.

[BC] = Name of the business component; for example, Account.

[CHILD_ID] = Child Record ID; for example, 42-6ODBL.

[PARENT_ID] = Parent Record ID; for example, 42-6ODBL.

- Establishes a 1:1 association. one parent record will be associated with one child record
- Multiple 1:1 associations should be made for a 1:N association and multiple 1:N associations should be made for a M:N association

Please note:

- The parent search expression must match a unique record in the parent table.
- The child search expression must match a unique record in the child table.

The return value contains the record ID of the associated child row and the record ID of the associated parent row. If the parent search expression or the child search expression returns more than one matching record, an exception will be thrown.

Di Dissociate message:

The Dissociate operation is surfaced onl

```
<Dissociate xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation">
  <ViewMode>[View_mode]</ViewMode>
  <ParentSearchExpr>Par_search</ParentSearchExpr>
  <ParentMVGField>Field_name</ParentMVGField>
  <ChildSearchExpr>Chld_search</ChildSearchExpr>
</Dissociate>
```

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[BO] = Name of the business object; for example, Account.

[BC] = Name of the business component; for example, Account.

[View_mode] = View mode to apply to the query for parent and child business component records. If a value is specified, that value will be applied to both the parent search and the child search. This is an optional argument. For an explanation of permissible values, see the Siebel documentation.

[Par_search] = Search expression matching exactly one parent business component record; for example, [Id] LIKE AB-12345.

[Chld_search] = Search expression matching exactly one child business component record that will be dissociated with the parent business component record; for example, [Id] LIKE CD-12345.

[Field_name] = Multi-value group field name of the parent business component; for example, 'Bill To First Name' in the Account business component.

Dissociate response message:

```
<DissociateResponse xmlns="[VERSION]/BusinessObjects/[BO]/[BC]/Operation">
  <DissociateResult>
    <ChildID>[CHILD_ID]</ChildID>
    <ParentID>[PARENT_ID]</ParentID>
  </DissociateResult>
</DisocciateResponse>
```

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

y for business components with multi-value group (MVG) fields. It dissociates records by specifying search expressions for parent and child records.

The Dissociate operation:

- Dissociates a selected parent business component record with one selected child business component record
- Establishes a 1:1 dissociation. one parent record will be dissociated with one child record
- Multiple 1:1 dissociations should be made for a 1:N dissociation and multiple 1:N dissociations should be made for a M:N dissociation

Note that the child search expression must match a unique record among the associated records with parent records that match the parent search expression.

◆ Important

There is a difference in the child search expression for ASSOCIATE and DISSOCIATE operations. In an ASSOCIATE operation, the child search expression searches all records in the child business component. In a DISSOCIATE operation, the child search expression searches only those records in the child business component that are associated with records in the parent business component, where the parent records match the parent search expression.

The return value contains the record ID of the dissociated child record and the record ID of the dissociated parent record. If the parent search expression or the child search expression returns more than one matching record, an exception will be thrown.

<p>ices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component ; for example, Account.</p> <p>[CHILD_ID] = Child Record ID; for example, 42-6ODBL.</p> <p>[PARENT_ID] = Parent Record ID; for example, 42-6ODBL.</p>	
<p>Query_[MVG_Child_Business_Comp] message:</p> <pre> <Query_[CHILD_BC] xmlns=" BusinessObjects/[BO]/[BC] /Operation"> <ViewMode>View_mode</ViewMode> <ParentSearchExpr>Par_search</ParentSearchExpr> <ParentMVGField>Field_name</ParentMVGField> <[CHILD_BC]QueryInputRecord> <SearchExpr>Chld_search </SearchExpr> <QueryFields> <string>field1</string> <string>field2</string> ... </QueryFields> </[CHILD_BC]QueryInputRecord> </Query_[CHILD_BC]> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[BO] = Name of the business object; for example, Account.</p> <p>[BC] = Name of the business component ; for example, Account.</p> <p>[View_mode] = View mode to apply; can be 0 to 9. This is an optional parameter. See the Siebel documentation for an explanation of these possible values.</p> <p>[CHILD_BC] = Name of the child business component associated with the parent business component's MVG field; for example, Contact (for the Account parent business component).</p> <p>[Par_search] = Search expression matching 1 parent business component record; for example, [Id] LIKE AB-12345.</p>	<p>This operation is exposed for business components with MVG fields.</p> <p>It queries the child records associated with a parent record that is associated with a MVG field in the parent business component.</p>

[Field_name] = Multi-value group field name of the Parent business component; for example, 'Bill To First Name' in the Account business component'.

[Chld_search] = Search expression matching 1 or more child business component records; for example, [Id] LIKE CD-12345.

[field1], [field2], and so on = Query fields in the child business component. A list of field names that specifies the fields in the child business component that should be retrieved by the operation. Only the fields in this list are returned in each record. Each field should be specified by using the original name of the field in the child business component, not the XML encoded name; for example, "First Name" instead of "First_x0032_Name". This parameter is optional.

Query_[MVG_Child_Business_Comp] response message:

```
<Query_[CHILD_BC]Response
xmlns="[VERSION]/Business
Objects/[BO]/[BC]/Operati
on">
  <Query_[CHILD_BC]Result
  >
    <[CHILD_BC]QueryRecor
d>
      <[FIELD1_NAME]>valu
e1</[FIELD1_NAME]>
      <[FIELD2_NAME]>valu
e2</[FIELD2_NAME]>
      ...
    </[CHILD_BC]QueryReco
rd >
  </QueryResult>
</QueryResponse>
```

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[BO] = Name of the business object; for example, Account.

[BC] = Name of the business component; for example, Account.

[FIELD1_NAME] = business component field name; for example, Account_x0020_Status.

[CHILD_BC] = Name of the child business component associated with the parent business component's MVG field; for example, Contact (for the Account parent business component).

The following table shows the SOAP action for each of the Siebel business component operations. Only the actions for the request messages are shown, the action for the response message is formed by appending "/response" to the request message action; for example, "http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert/response".

Operation	Action	Example
Insert	[VERSION]/BusinessObjects/[BO]/[BC]/Insert	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert
Query	[VERSION]/BusinessObjects/[BO]/[BC]/Query	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Query
Update	[VERSION]/BusinessObjects/[BO]/[BC]/Update	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Update
Delete	[VERSION]/BusinessObjects/[BO]/[BC]/Delete	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Delete
Associate	[VERSION]/BusinessObjects/[BO]/[BC]/Associate	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Associate
Dissociate	[VERSION]/BusinessObjects/[BO]/[BC]/Dissociate	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Dissociate
Query_[CHILD_BC]	[VERSION]/BusinessObjects/[BO]/[BC]/Query_[CHILD_BC]	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Query_Contact

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[BO] = Business object name; for example, Account.

[BC] = Business component name; for example, Account.

Name of the child business component associated with the parent business component's MVG field; for example, Contact (for the Account parent business component).

You must explicitly specify the message action when you consume the Siebel adapter in a BizTalk Server solution or by using the Windows Communication Foundation (WCF) channel model. For more information, see [Development](#).

Siebel Business Component WCF Client Methods

The following table shows the WCF service model method signatures that are generated by the Add Adapter Service Reference Visual Studio Plug-in for Siebel business components.

Operation	WCF Service Model Method
Insert	<code>string[] client.Insert([BC]InsertRecord[] ArrayOf[BC]InsertRecord);</code>
	[BC] = Name of business component; for example, Account.
Query	<code>[BC]QueryRecord[] client.Query(System.Nullable<short> ViewMode, [BC]QueryInputRecord [BC]QueryInputRecord);</code>
	[BC] = Name of business component, for example, Account.
Update	<code>string[] client.Update(System.Nullable<short> ViewMode, [BC]UpdateRecord[] ArrayOf[BC]UpdateRecord);</code>
	[BC] = Name of business component; for example, Account.

Delete	<code>string[] client.Delete(System.Nullable<short> ViewMode, string[] Id, string SearchExpr);</code>
Associate	<code>ParentChildRecord client.Associate(System.Nullable<short> ViewMode, string ParentSearchExpr, string ParentMVGField, string ChildSearchExpr);</code>
Dissociate	<code>ParentChildRecord client.Dissociate(System.Nullable<short> ViewMode, string ParentSearchExpr, string ParentMVGField, string ChildSearchExpr);</code>
Query_[MVG Child Business Comp]	<code>[CHILD_BC]QueryRecord[] client.Query_[CHILD_BC](System.Nullable<short> ViewMode short ViewMode, string ParentSearchExpr, string ParentMVGField, [CHILD_BC]QueryInputRecord [CHILD_BC]QueryInputRecord);</code>
	[CHILD_BC] = Name of the child business component associated with the parent business component's MVG field; for example, Contact (for the Account parent business component).

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)

Message Schemas for Business Service Operations

A Siebel business service is a collection of business methods that can be directly invoked on a Siebel system. The Siebel adapter surfaces the business methods of a Siebel business service as operations.

Message Schemas for Siebel Business Service Method Operations

The following table shows the message schemas for Siebel business service method operations surfaced by the Siebel adapter.

Operation	XML Structure	Description
-----------	---------------	-------------

<p>[Business Service Method Name]</p>	<p>Business Service Method Request message:</p> <pre> <[METHOD_NAME] xmlns="[VERSION]/BusinessServices/[Business Service]/Operation"> <[METHOD_NAME]RequestRecord> <[I_PRM1_NAME]>value1</[I_PRM1_NAME]> <[I_PRM2_NAME]>value2</[I_PRM2_NAME]> ... </[METHOD_NAME]RequestRecord> <[METHOD_NAME]InOutRecord> <[IO_PRM1_NAME]>value1</[IO_PRM1_NAME]> <[IO_PRM2_NAME]>value2</[IO_PRM2_NAME]> ... </[METHOD_NAME]InOutRecord> </[METHOD_NAME]> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[Business Service] = The name of business service; for example, ExtractDataService.</p> <p>[METHOD_NAME] = The name of the business service method; for example, ExecuteNext.</p> <p>[I_PRM_NAME] = Names of IN parameters.</p> <p>[IO_PRM_NAME] = Names of IN OUT parameters.</p> <p>Business Service Method Response message:</p> <pre> <[METHOD_NAME]Response xmlns="[VERSION]/BusinessServices/[Business Service]/Operation"> <[METHOD_NAME]Result> <[O_PRM1_NAME]>value1</[O_PRM1_NAME]> <[O_PRM2_NAME]>value2</[O_PRM2_NAME]> ... </[METHOD_NAME]Result> <[METHOD_NAME]InOutRecord> <[IO_PRM1_NAME]>value1</[IO_PRM1_NAME]> <[IO_PRM2_NAME]>value2</[IO_PRM2_NAME]> ... </[METHOD_NAME]InOutRecord > </[METHOD_NAME]Response> </pre> <p>[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".</p> <p>[Business Service] = Name of business service; for example, ExtractDataService.</p> <p>[METHOD_NAME] = Name of the business service method; for example, ExecuteNext.</p> <p>[O_PRM_NAME] = Names of OUT parameters.</p> <p>[IO_PRM_NAME] = Names of INOUT parameters.</p> <p>◆ Important The IN OUT and OUT parameters are always marked as optional in the metadata, even if they required by the Siebel system. So, if a parameter is marked as optional in the metadata but required by the Siebel system, the adapter throws the <code>TargetSystemException</code> as received from Siebel and not the <code>XmlReaderParsingException</code>.</p>	<p>The Siebel business service method is surfaced as an operation name.</p> <ul style="list-style-type: none"> • IN, IN OUT and OUT parameters are supported. • Hierarchical types are surfaced as strings. The Siebel adapter does not validate the values passed for these strings. If these values do not conform to the schemas expected by the Siebel system, a run-time exception will be generated.
---------------------------------------	---	--

Message Actions for Siebel Business Service Method Operations

The following table shows how the SOAP action for a Siebel Business Service method is formed. Only the action for the request message is shown, the action for the response message is formed by appending "/response" to the request message action; for example, "[VERSION]/BusinessServices/ExtractDataService/ExecuteNext/response".

Operation	Action	Description
[Business_Service_METHOD_NAME]	[VERSION]/BusinessServices/[Business Service]/[Business_Service_METHOD_NAME]	[VERSION]/BusinessServices/ExtractDataService/ExecuteNext

[VERSION] = The message version string; for example, "http://Microsoft.LobServices.Siebel/2007/03".

[Business Service] = The name of business service; for example, ExtractDataService.

[Business_Service_METHOD_NAME] = The name of the business service method; for example, ExecuteNext.

You must explicitly specify the message action when you consume the Siebel adapter in a BizTalk Server solution or by using the WCF channel model. For more information, see [Development](#).

Siebel Business Service WCF Client Methods

The following table shows the WCF service model method signature that is generated by the Add Adapter Service Reference Visual Studio Plug-in for Siebel business services methods.

Operation	WCF Service Model Method
[Business_Service_METHOD_NAME]	[Business_Service_METHOD_NAME]ResponseRecord client.[Business_Service_METHOD_NAME]([Business_Service_METHOD_NAME]RequestRecord);
	[Business_Service_METHOD_NAME] = Business service method name; for example, ExecuteNext.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)

Message Schema for Picklist Operations

Picklists are special field types in business components. They represent a collection of values from which a user may pick a single value for assignment. Picklists are of different types. For more information about picklists and their types, refer to Siebel documentation.

One of the picklist types, static bounded picklists, are surfaced by the adapters as an enumerated picklist type in the metadata generated by the adapter at design time. This contains a static set of values that must be specified for the picklist type at run time. While specifying a value for a static bounded picklist, you must always specify a value that belongs to the set.

The following sample shows the schema of a static bounded picklist type:

```
<element name="[FIELD_NAME]RequiredPickListType" nillable="true" type="ns1:[FIELD_NAME]RequiredPickListType" />
<simpleType name="[FIELD_NAME]RequiredPickListType">
  <restriction base="string">
    <enumeration value="value1" />
    <enumeration value="value2" />
    ...
  </restriction>
</simpleType>
```

[FIELD_NAME] = Picklist field name in the business component

The following represents the proxy experience of a static bounded picklist type:

```
[BC]InsertRecord[] insertRecs = new [BC]InsertRecord[1];
insertRecs[0] = new [BC]InsertRecord();
insertRecs[0].[BC_STATIC_PICKLIST_FIELD] = [BC_PICKLIST_FIELD_NAME]OptionalPickListType.value1;
```

[BC_STATIC_PICKLIST_FIELD] = A static bounded picklist field in the BC

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)

Message Versioning Support

The Microsoft BizTalk Adapter for Siebel eBusiness Applications supports versioning by including a version string component in the message actions, namespaces, and node IDs surfaced for operations. The current version is <http://Microsoft.LobServices.Siebel/2007/03>. This means that for an Insert operation on an Account business object in the Siebel repository, the Insert operation surfaced by the adapter has the following:

- Node ID: <http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert>
- Message action: <http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert>
- Namespace: <http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Operation>

 **Note**

This feature does not provide backward compatibility with the earlier versions of the adapter.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#)

Metadata Node IDs

Siebel Node IDs

The Siebel adapter surfaces Siebel system artifacts in a hierarchical manner. The following table describes the node types and node IDs for Siebel artifacts surfaced by the Siebel adapter.

Artifact	Node Type	Node Id	Example	Description
Business Objects (All Business Objects)	CATEGORY	[VERSION]/BusinessObjects	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects	Returns all business objects.
Business Object (BO)	CATEGORY	[VERSION]/BusinessObjects/[BO]	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account	Returns all business components associated with the specified business object.
Business Component (BC)	CATEGORY	[VERSION]/BusinessObjects/[BO]/[BC]	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account	Returns all operations associated with the specified business component.
Insert	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Insert	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert	Returns the Insert operation for the specified business component.
Query	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Query	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Query	Returns the Query operation for the specified business component.
Update	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Update	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Update	Returns the Update operation for the specified business component.
Delete	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Delete	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Delete	Returns the Delete operation for the specified business component.
Associate	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Associate	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Associate	Returns the Associate operation for the specified business component.
Dissociate	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Dissociate	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Dissociate	Returns the Dissociate operation for the specified business component.
Query_[MVG Child Business Component]	OPERATION	[VERSION]/BusinessObjects/[BO]/[BC]/Query_[MVG Child Business Component]	http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Query_Contact	Returns the Query operation for the child business component.
Business Services	CATEGORY	[VERSION]/BusinessServices	http://Microsoft.LobServices.Siebel/2007/03/BusinessServices	Returns all business services.
Business Service	CATEGORY	[VERSION]/BusinessServices/[Business Service]	http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/ATP	Returns all business methods for the specified business service.

Business Service Method	OPERATION	[VERSION]/BusinessServices/[Business Service Method]	http://Microsoft.LobServices.Siebel/2007/03/BusinessServices/ATP/ATPRunCheck	Returns the specified business service method.
-------------------------	-----------	--	--	--

[VERSION] = The version string; for example http://Microsoft.LobServices.Siebel/2007/03.

[BO] = A Siebel business object; for example, Account.

[BC] = A business component; for example, Account.

[Business Service] = A Siebel business service; for example, ATP.

[Business Service Method] = A method of a business service; for example, DismissAlarm.

[MVG Child Business Component] = A multi-value group child business component; for example, Contact.

Metadata Search and Node IDs

Metadata search is a powerful feature that the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK surfaces as part of its **MetadataRetrievalContract** interface. The Siebel adapter leverages this feature to support searching the following Siebel artifacts.

Artifact	Node Id	Description
Business Object	[VERSION]/BusinessObjects	Return business objects that match the search expression.
Business Component	[VERSION]/BusinessObjects/[BO]	Return business components that match the search expression.
Business Service	[VERSION]/BusinessServices	Return business services that match the search expression.
Business Service Method	[VERSION]/BusinessServices/[Business Service]	Return business service methods that match the search expression.

[VERSION] = The version string; for example http://Microsoft.LobServices.Siebel/2007/03.

[BO] = A Siebel Business Object; for example, Account.

[Business Service] = A Siebel Business Service; for example, ATP.

For valid search expressions, refer to the Siebel documentation.

Note
The Siebel adapter only supports searches at the level below the currently selected node. For example, when BusinessObjects is selected, A* is a supported search, but A*/A* is not.

Metadata Retrieval and Node IDs

The Siebel adapter captures the following characteristics for each type of artifact.

Artifact	Metadata Characteristics
----------	--------------------------

Business Component	<ul style="list-style-type: none"> • Business component name • Business component field names • Business component field data types mapped to simple/complex WSDL types • Business component field length mapped to facet maxLength • Business component mandatory field mapped to facet minOccurs = 1 • Business component optional field mapped to facet minOccurs = 0 • Business component picklist field mapped to a Picklist complex type in WSDL • Business component static bounded picklist to contain an enumerated list of values • Business component field NULL constraint mapped to facet isNillable = true • Business component operations <ul style="list-style-type: none"> • INSERT • QUERY • UPDATE • DELETE • ASSOCIATE • DISSOCIATE • QUERY_[MVG Child Business Comp] for each child business component with which the business component has an m:n relationship
Business Service Method	<ul style="list-style-type: none"> • Business Service name • Method name • Method is the operation • Method parameter names • Method parameter data types mapped to WSDL types • Method parameter direction mapped to WSDL parameter direction • Method parameter order mapped to element sequence

For detailed information about the format of the metadata that the Siebel adapter exposes for specific artifacts and operations on the Siebel system, see [Messages and Message Schemas for BizTalk Adapter for Siebel eBusiness Applications](#).

See Also
Other Resources

Using the ServiceModel Metadata Utility Tool with the BizTalk Adapter for Siebel eBusiness Applications

You can use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate a WCF client class for operations that the Microsoft BizTalk Adapter for Siebel eBusiness Applications exposes. After you run svcutil.exe to generate a WCF client class, you can include the generated file in your code and create instances of the WCF client class to perform operations on the Siebel system.

Using svcutil.exe requires you to supply a connection URI that contains credentials. Because, by default, the Siebel adapter disables credentials in the connection URI, you must configure svcutil.exe to use a non-default binding for the Siebel adapter. You can also configure other binding properties in the non-default binding.

The following sections show you how to configure svcutil.exe and how to use svcutil.exe to generate WCF client code with the Siebel adapter.

Configuring svcutil.exe for the Siebel Adapter

To configure svcutil.exe to use a non-default binding, you must create a local copy of svcutil.exe and then create or modify a local copy of the svcutil.exe.config configuration file.

To configure svcutil.exe for a non-default binding

1. Create a folder, and copy svcutil.exe into the new folder. You can typically find svcutil.exe at the Windows SDK installation location, specifically, C:\Program Files\Microsoft SDKs\Windows\v6.0\Bin.
2. Create a file named svcutil.exe.config in the new folder.
3. Add a binding and a client endpoint to the svcutil.exe.config file. You must run svcutil.exe from the new folder to ensure that the correct configuration is used.

Important

The name attribute of the client endpoint must specify the scheme used in the connection URI. This value is case-sensitive.

```
<configuration>
  <system.serviceModel>
    <client>
      <!-- the name should match the required scheme of the Metadata Exchange endpoint
      and the contract should be "IMetadataExchange" -->
      <endpoint name="siebel"
        binding="siebelBinding"
        bindingConfiguration="SiebelBinding"
        contract="IMetadataExchange" />
    </client>
    <bindings>
      <siebelBinding>
        <binding name="SiebelBinding" acceptCredentialsInUri="true" />
      </siebelBinding>
    </bindings>
  </system.serviceModel>
</configuration>
```

Note

You can set any of the binding properties of the Siebel adapter in the binding configuration.

For more information about configuring a non-default binding for svcutil.exe, see the "Custom Secure Metadata Endpoint" topic in the WCF documentation at <http://go.microsoft.com/fwlink/?LinkId=96077>.

Creating a WCF Client Class with svcutil.exe

To use svcutil.exe to generate WCF client code for the Siebel adapter, you must supply a connection URI that specifies an **IMetadataExchange** (mex) endpoint and the operation or operations for which you want svcutil.exe to generate code. You must also specify connection credentials for the Siebel system in the connection URI.

Note

Before you can use svcutil.exe with the Siebel adapter, you must configure it to use a non-default binding; for information about how to do this, see [Configuring svcutil.exe for the Siebel Adapter](#).

You specify a mex endpoint and target operations in the Siebel adapter connection URI in the following manner:

- You must include the "wsdl" parameter in the query_string. If it is the first parameter in the query_string, it is specified just after the question mark (?). If it is not the first parameter, it should be preceded with an ampersand (&).
- You must follow the "wsdl" parameter by one or more "op" parameters. Each "op" parameter is preceded by an ampersand (&) and specifies the node ID of a target operation.

The following two examples show how to target various operations by using svcutil.exe.

This example creates a WCF client class for an insert operation on the ACCOUNT\ACCOUNT Business Object.

```
.\svcutil "siebel://Username=YourUserName;Password=YourPassword@Siebel_server:1234?
SiebelEnterpriseServer=ent_server&SiebelObjectManager=obj_mgr&Language=enu&wsdl&op=http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert"
```

This example creates a WCF client class for both an insert operation and a delete operation on the ACCOUNT\ACCOUNT Business Object.

```
.\svcutil " siebel://Username=YourUserName;Password=YourPassword@Siebel_server:1234?
SiebelEnterpriseServer=ent_server&SiebelObjectManager=obj_mgr&Language=enu&wsdl&op=http://Microsoft.LobServices.Siebel/2007/03/BusinessObjects/Account/Account/Insert&op
```

Important

You must place the connection URI in quotation marks on the command line. Otherwise, svcutil.exe attempts to retrieve metadata for operations that the Siebel adapter does not support. The results of such an attempt are undefined.

By default, svcutil.exe places the generated code in the output.cs file; however, you can change the name of the output file and many other options that svcutil.exe uses by setting command-line switches.

Svcutil.exe does not provide the capability to search for operations (for example, by using wildcard characters). You must explicitly specify node IDs for the specific operations you want to target. You cannot specify node IDs that refer only to categories. For more information about the node IDs that the Siebel adapter surfaces, see [Metadata Node IDs](#).

The Add Adapter Service Reference Visual Studio Plug-in provides advanced browse and search capabilities that can greatly simplify generating a WCF client class. For more information about the Add Adapter Service Reference Plug-in, see [Generating a WCF Client for Siebel Artifacts](#).

See Also

Other Resources

[Technical Reference](#)

Glossary

The following terms and definitions are used in Microsoft BizTalk Adapter for Siebel eBusiness Applications Help.

Microsoft BizTalk Adapter Pack Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

adapter	A WCF-based component that helps exchange messages between applications (for example, a line-of-business system) and BizTalk Server. The adapter consists of design-time components and run-time components for receive and send operations.
adapter client	Refers to an application that is interacting with a line-of-business (LOB) system through the adapter.

B

binding	A process by which software components and layers are linked together. When a network component is installed, the binding relationships and dependencies for the components are established. Binding allows components to communicate with each other. In BizTalk Server, an established mapping between an orchestration adapter-agnostic endpoint (port or role link) and physical adapter-specific endpoints (send/receive ports or party).
BizTalk Server	Connects diverse software. BizTalk Server enables you to create and modify process logic that uses that software. BizTalk Server also enables information workers to monitor running processes, interact with trading partners, and perform other business-oriented tasks.

C

channel	A concrete implementation of a binding element. The binding represents the configuration, and the channel is the implementation associated with that configuration. Therefore, there is a channel associated with each binding element. Channels stack on top of each other to create the concrete implementation of the binding: the channel stack.
connection URI	A string that identifies a resource in a distributed environment. Adapters use a connection Uniform Resource Indicator (URI) that contains the information necessary to establish a connection with the LOB system.
contract	Specifies the collection and structure of messages required to access the operations offered by the service.

D

design-time experience	Procedures and operations that a developer performs during design time; for example, using the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas.
------------------------	--

E

endpoint address	A network address that identifies the location of a Windows Communication Foundation (WCF) service endpoint. For an adapter, the endpoint address is expressed as a connection Uniform Resource Identifier (URI) that contains location and connection parameters. The adapter can use these to establish a connection to the underlying line-of-business (LOB) system.
Enterprise Single Sign-on system	An SSO database, a master secret server, and one or more Enterprise Single Sign-On (SSO) servers. These servers do the mapping between the Windows and non-Windows credentials, look up the credentials in the SSO database, and are used for administering the SSO system. The SSO database is also used as a configuration store to hold custom configuration data for adapters.

Extensible Markup Language	A markup language designed to describe data. XML tags are not predefined.
----------------------------	---

G

GAC	See global assembly cache.
global assembly cache (GAC)	A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer. Applications deployed in the global assembly cache must have a strong name.

H

Health and Activity Tracking (HAT)	A user interface that enables specified data to be tracked. This interface can be used to monitor the health of the BizTalk server farm and to track and view specific events and messages.
------------------------------------	---

I

inbound operation	An operation that is invoked by a line-of-business (LOB) system on the adapter.
-------------------	---

M

metadata	Information, such as location, time, message size, and/or exception information.
Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK	The specifications for building BizTalk adapters using open standards based on Web services.

O

one-way	A message exchange pattern (MEP) in which the sender sends a message, but no response is returned by the receiver. In BizTalk Server, MEPs are referred to as communication patterns.
outbound operation	An operation that is invoked by the adapter on the line-of-business system (LOB).
output.cs	The default output file created by the ServiceModel Metadata Utility tool (svcutil.exe).

P

picklist	A field in a business component whose value is typically constrained to one of a set of values (akin to enumeration).
proxy	In WCF, refers to a managed-code object that implements the service contract exposed by a service. The WCF service model is based on the use of such proxies. In the WCF service model, the service contract is expressed as a .NET interface.

R

request-reply	A message exchange pattern (MEP) in which the sender sends a request message and expects a response message from the receiver. In BizTalk Server, MEPs are referred to as communication patterns. Depending on the messaging technology and the direction of the request message (inbound or outbound), this pattern is also called request-reply or solicit-response.
runtime experience	Procedures and operations performed by a developer during run time or when deploying a solution; for example, creating a physical port binding from the BizTalk Server Administration console.

S

schema	The structure for a message. A schema can contain multiple subschema.
--------	---

ServiceModel Metadata Utility tool	A command-line utility that is included with WCF. It is used to create service model proxy code from the service description (metadata) that is exposed by a WCF service such as an adapter. For outbound operations, the tool creates a WCF client class and helper code; for inbound operations, the tool creates a WCF service contract and helper code.
Siebel business component	Associates logically related columns from one or more tables into a single structure.
Siebel business object	Represents a logical grouping of business components.
Siebel business service	A collection of business service methods or functions that can be directly invoked in the Siebel system.
Siebel COM Data Control library	Contains interfaces that enable an external client like the Siebel adapter to connect and communicate with a Siebel Application Object Manager on a Siebel server.
Simple Object Access Protocol	See SOAP.
SOAP	A simple, XML-based protocol for exchanging structured and type information in decentralized, distributed environments. WCF is based on the exchange of SOAP messages between clients and services to invoke operations and return results.
SOAP message	A well-formed XML document. It should use the SOAP envelope and SOAP encoding namespaces and include an optional XML declaration, followed by a SOAP envelope (the root element), which is made up of an optional SOAP header and a SOAP message body.
SQL Server Integration Services	A component that is used to import, export, and transform data from different data sources. Previously called data transformation service (DTS).
SSIS	See SQL Server Integration Services.
SSO	See Enterprise Single Sign-on system.
strongly-typed data	A data set or result set that is bound to an underlying object type. Each row in a strongly-typed XML data set is composed of typed, named elements that correspond to fields of the underlying object type.
svcutil	See ServiceModel Metadata Utility tool.

U

Uniform Resource Identifier (URI)	See connection URI.
-----------------------------------	---------------------

W

WCF	See Windows Communication Foundation.
WCF channel model	A programming model that relies on several interfaces and other types. Channels provide a low-level programming model for sending and receiving messages.

WCF client	A client-application construct that exposes the service operations as methods. You can use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool to generate a WCF client class from the metadata exposed by an adapter.
WCF LOB Adapter SDK	See Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
WCF service contract	A managed-code representation of the service contract. It is expressed as an interface in which classes and methods are attributed to define the service, operation, message, and data contracts used to communicate with a service. You can use the ServiceModel Metadata Utility tool or the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF service contract from the metadata exposed by an adapter. You implement the WCF service contract to receive operations from an LOB system.
WCF service model	A WCF programming model in which a service is represented as a managed code object. The operations exposed by the service are represented as methods with strongly-typed data.
weakly-typed data	A data set or result set that is not bound to an underlying object type. Each row in a weakly-typed XML data set is composed of a collection of generic columns in which attributes describe the name and type of each element.
Web services	A unit of application logic providing data and services to other applications. Applications access XML Web services using standard Web protocols and data formats such as HTTP, XML, and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web , and are a cornerstone of the Microsoft .NET programming model.
Web Services Description Language	An XML-based language that describes a service as a set of endpoints that operate on messages. The WSDL document describes the service contract, operation contracts, message contracts, and data contracts that a client must use to interface with the service.
Windows Communication Foundation (WCF)	A Microsoft service-oriented communication infrastructure. The framework inherently provides clients with a service programming model and a channel programming model for finer control of message exchanges.
WSDL	See Web Services Description Language.
WS-Metadata Exchange (MEX) endpoint	An endpoint exposed by a WCF service, such as an adapter, that implements the IMetadataExchange interface. A WS-Metadata Exchange endpoint can be used to retrieve a service description (WSDL) for operations exposed by an adapter on the target system.

X

XML	See Extensible Markup Language.
XML Schema definition language (XSD)	A schema language. An XML Schema defines the elements, attributes, and data types that comply with the World Wide Web Consortium (W3C) XML Schema Part 1: Structures Recommendation for the XML Schema Definition Language. The W3C XML Schema Part 2: Datatypes Recommendation is the recommendation for defining data types that are used in XML schemas. The XML Schema definition language enables you to define the structure and data types for XML messages.
XSD	See XML Schema definition language.

Class Library

Namespaces

Namespace
Microsoft.Adapters.Siebel
Microsoft.Data.SiebelClient

Microsoft.Adapters.Siebel Namespace

Classes

Class	Description
BindingConfigConstants	
BindingDefaultConstants	
SiebelAdapter	
SiebelAdapterExtensionElement	
SiebelBinding	
SiebelBindingConfigElement	
SiebelConnectionUri	
SiebelConstants	

BindingConfigConstants Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class BindingConfigConstants
```

C#

```
public static class BindingConfigConstants
```

C++

```
public ref class BindingConfigConstants abstract sealed
```

J#

```
public final class BindingConfigConstants
```

JScript

```
public final class BindingConfigConstants
```

Inheritance Hierarchy [System.Object](#)

Microsoft.Adapters.Siebel.BindingConfigConstants

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants Members

The following tables list the members exposed by the [BindingConfigConstants](#) type.

Public Fields

	Name	Description
	AcceptCredentialsInUri	
	BatchFields	
	ConnectionCategory	
	DiagnosticsCategory	
	EnableBizTalkCompatibilityMode	
	EnableConnectionPooling	
	EnablePerformanceCounters	
	IdleConnectionTimeout	
	LogData	
	MaxConnectionsPerSystem	
	MiscellaneousCategory	
	PerformanceCategory	

Top

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Top

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[BindingConfigConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants Fields

Public Fields

	Name	Description
 	AcceptCredentialsInUri	
 	BatchFields	
 	ConnectionCategory	
 	DiagnosticsCategory	
 	EnableBizTalkCompatibilityMode	
 	EnableConnectionPooling	
 	EnablePerformanceCounters	
 	IdleConnectionTimeout	
 	LogData	
 	MaxConnectionsPerSystem	
 	MiscellaneousCategory	
 	PerformanceCategory	

See Also

Reference

[BindingConfigConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.AcceptCredentialsInUri Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const AcceptCredentialsInUri As String
```

C#

```
public const string AcceptCredentialsInUri
```

C++

```
public:  
literal String^ AcceptCredentialsInUri
```

J#

```
public static final String AcceptCredentialsInUri
```

JScript

```
public const var AcceptCredentialsInUri : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.BatchFields Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const BatchFields As String
```

C#

```
public const string BatchFields
```

C++

```
public:  
literal String^ BatchFields
```

J#

```
public static final String BatchFields
```

JScript

```
public const var BatchFields : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.ConnectionCategory Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const ConnectionCategory As String
```

C#

```
public const string ConnectionCategory
```

C++

```
public:  
literal String^ ConnectionCategory
```

J#

```
public static final String ConnectionCategory
```

JScript

```
public const var ConnectionCategory : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.DiagnosticsCategory Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const DiagnosticsCategory As String
```

C#

```
public const string DiagnosticsCategory
```

C++

```
public:  
literal String^ DiagnosticsCategory
```

J#

```
public static final String DiagnosticsCategory
```

JScript

```
public const var DiagnosticsCategory : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.EnableBizTalkCompatibilityMode Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const EnableBizTalkCompatibilityMode As String
```

C#

```
public const string EnableBizTalkCompatibilityMode
```

C++

```
public:  
literal String^ EnableBizTalkCompatibilityMode
```

J#

```
public static final String EnableBizTalkCompatibilityMode
```

JScript

```
public const var EnableBizTalkCompatibilityMode : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.EnableConnectionPooling Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const EnableConnectionPooling As String
```

C#

```
public const string EnableConnectionPooling
```

C++

```
public:  
literal String^ EnableConnectionPooling
```

J#

```
public static final String EnableConnectionPooling
```

JScript

```
public const var EnableConnectionPooling : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.EnablePerformanceCounters Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId:="Perf")> _
Public Const EnablePerformanceCounters As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")]
public const string EnablePerformanceCounters
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId=L"Perf")]
public:
    literal String^ EnablePerformanceCounters
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly", MessageId="Perf") */
public static final String EnablePerformanceCounters
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")
public const var EnablePerformanceCounters : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.IdleConnectionTimeout Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const IdleConnectionTimeout As String
```

C#

```
public const string IdleConnectionTimeout
```

C++

```
public:  
literal String^ IdleConnectionTimeout
```

J#

```
public static final String IdleConnectionTimeout
```

JScript

```
public const var IdleConnectionTimeout : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.LogData Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const LogData As String
```

C#

```
public const string LogData
```

C++

```
public:  
literal String^ LogData
```

J#

```
public static final String LogData
```

JScript

```
public const var LogData : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.MaxConnectionsPerSystem Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const MaxConnectionsPerSystem As String
```

C#

```
public const string MaxConnectionsPerSystem
```

C++

```
public:  
literal String^ MaxConnectionsPerSystem
```

J#

```
public static final String MaxConnectionsPerSystem
```

JScript

```
public const var MaxConnectionsPerSystem : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.MiscellaneousCategory Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const MiscellaneousCategory As String
```

C#

```
public const string MiscellaneousCategory
```

C++

```
public:  
literal String^ MiscellaneousCategory
```

J#

```
public static final String MiscellaneousCategory
```

JScript

```
public const var MiscellaneousCategory : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants.PerformanceCategory Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const PerformanceCategory As String
```

C#

```
public const string PerformanceCategory
```

C++

```
public:  
literal String^ PerformanceCategory
```

J#

```
public static final String PerformanceCategory
```

JScript

```
public const var PerformanceCategory : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingConfigConstants Class](#)

[BindingConfigConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingConfigConstants Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[BindingConfigConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class BindingDefaultConstants
```

C#

```
public static class BindingDefaultConstants
```

C++

```
public ref class BindingDefaultConstants abstract sealed
```

J#

```
public final class BindingDefaultConstants
```

JScript

```
public final class BindingDefaultConstants
```

Inheritance Hierarchy [System.Object](#)

Microsoft.Adapters.Siebel.BindingDefaultConstants

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants Members

The following tables list the members exposed by the [BindingDefaultConstants](#) type.

Public Fields

	Name	Description
	AcceptCredentialsInUri	
	BatchFields	
	EnableCommonCache	
	EnableConnectionPool	
	EnablePerfCounters	
	HandlersShareSameConnection	
	IdleConnectionTimeout	
	LogData	
	MaxConnectionsPerSystem	

Top

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Top

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[BindingDefaultConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants Fields

Public Fields

	Name	Description
 	AcceptCredentialsInUri	
 	BatchFields	
 	EnableCommonCache	
 	EnableConnectionPool	
 	EnablePerfCounters	
 	HandlersShareSameConnection	
 	IdleConnectionTimeout	
 	LogData	
 	MaxConnectionsPerSystem	

See Also

Reference

[BindingDefaultConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.AcceptCredentialsInUri Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const AcceptCredentialsInUri As Boolean
```

C#

```
public const bool AcceptCredentialsInUri
```

C++

```
public:  
literal bool AcceptCredentialsInUri
```

J#

```
public static final boolean AcceptCredentialsInUri
```

JScript

```
public const var AcceptCredentialsInUri : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.BatchFields Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const BatchFields As Boolean
```

C#

```
public const bool BatchFields
```

C++

```
public:  
literal bool BatchFields
```

J#

```
public static final boolean BatchFields
```

JScript

```
public const var BatchFields : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.EnableCommonCache Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Const EnableCommonCache As Boolean
```

C#

```
public const bool EnableCommonCache
```

C++

```
public:  
literal bool EnableCommonCache
```

J#

```
public static final boolean EnableCommonCache
```

JScript

```
public const var EnableCommonCache : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.EnableConnectionPool Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const EnableConnectionPool As Boolean
```

C#

```
public const bool EnableConnectionPool
```

C++

```
public:  
literal bool EnableConnectionPool
```

J#

```
public static final boolean EnableConnectionPool
```

JScript

```
public const var EnableConnectionPool : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.EnablePerfCounters Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId:="Perf")> _
Public Const EnablePerfCounters As Boolean
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")]
public const bool EnablePerfCounters
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId=L"Perf")]
public:
    literal bool EnablePerfCounters
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly", MessageId="Perf") */
public static final boolean EnablePerfCounters
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")
public const var EnablePerfCounters : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.HandlersShareSameConnection Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const HandlersShareSameConnection As Boolean
```

C#

```
public const bool HandlersShareSameConnection
```

C++

```
public:  
literal bool HandlersShareSameConnection
```

J#

```
public static final boolean HandlersShareSameConnection
```

JScript

```
public const var HandlersShareSameConnection : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.IdleConnectionTimeout Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const IdleConnectionTimeout As Double
```

C#

```
public const double IdleConnectionTimeout
```

C++

```
public:  
literal double IdleConnectionTimeout
```

J#

```
public static final double IdleConnectionTimeout
```

JScript

```
public const var IdleConnectionTimeout : double
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.LogData Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Const LogData As Boolean
```

C#

```
public const bool LogData
```

C++

```
public:  
literal bool LogData
```

J#

```
public static final boolean LogData
```

JScript

```
public const var LogData : boolean
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants.MaxConnectionsPerSystem Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Const MaxConnectionsPerSystem As Integer
```

C#

```
public const int MaxConnectionsPerSystem
```

C++

```
public:  
literal int MaxConnectionsPerSystem
```

J#

```
public static final int MaxConnectionsPerSystem
```

JScript

```
public const var MaxConnectionsPerSystem : int
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[BindingDefaultConstants Class](#)

[BindingDefaultConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

BindingDefaultConstants Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[BindingDefaultConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapter Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelAdapter  
    Inherits CommonAdapter
```

C#

```
public sealed class SiebelAdapter : CommonAdapter
```

C++

```
public ref class SiebelAdapter sealed : public CommonAdapter
```

J#

```
public final class SiebelAdapter extends CommonAdapter
```

JScript

```
public final class SiebelAdapter extends CommonAdapter
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.BindingElement](#)

[System.ServiceModel.Channels.TransportBindingElement](#)

[Microsoft.ServiceModel.Channels.Common.Adapter](#)

[Microsoft.Adapters.AdapterUtilities.CommonAdapter](#)

Microsoft.Adapters.Siebel.SiebelAdapter

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapter Members

The following tables list the members exposed by the [SiebelAdapter](#) type.

Public Constructors

Name	Description
SiebelAdapter	

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 BatchFields	
 EnableConnectionPooling	
 EnablePerformanceCounters	
 IdleConnectionTimeout	
 LogData	
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxConnectionsPerSystem	
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 Scheme	Overridden.
 TransactedReceiveEnabled	(inherited from Adapter)

Top

Protected Properties

Name	Description
 Namespace	(inherited from Adapter)
 Settings	(inherited from Adapter)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 BuildChannelFactory	(inherited from Adapter)
 BuildChannelListener	(inherited from Adapter)
 CanBuildChannelFactory	(inherited from Adapter)
 CanBuildChannelListener	(inherited from Adapter)
 Clone	(inherited from Adapter)
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetProperty	(inherited from TransportBindingElement)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 BuildConnectionFactory	(inherited from Adapter)
 BuildConnectionUri	(inherited from Adapter)
 BuildWsdRetrieval	(inherited from Adapter)
 CloneAdapter	(inherited from Adapter)
 Finalize	(inherited from Object)
 IsHandlerSupported	(inherited from Adapter)
 MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Constructor

Initializes a new instance of the [SiebelAdapter](#) Class.

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelAdapter ()
```

C++

```
public:  
SiebelAdapter ()
```

J#

```
public SiebelAdapter ()
```

JScript

```
public function SiebelAdapter ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

[SiebelAdapter Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapter Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	BuildConnectionFactory	(inherited from Adapter)
	BuildConnectionUri	(inherited from Adapter)
	BuildWsdRetrieval	(inherited from Adapter)
	CloneAdapter	(inherited from Adapter)
	Finalize	(inherited from Object)
	IsHandlerSupported	(inherited from Adapter)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelAdapter Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapter Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	AcceptCredentialsInUri	
	BatchFields	
	EnableConnectionPooling	
	EnablePerformanceCounters	
	IdleConnectionTimeout	
	LogData	
	ManualAddressing	(inherited from TransportBindingElement)
	MaxBufferPoolSize	(inherited from TransportBindingElement)
	MaxConnectionsPerSystem	
	MaxReceivedMessageSize	(inherited from TransportBindingElement)
	Scheme	Overridden.
	TransactedReceiveEnabled	(inherited from Adapter)

Protected Properties

	Name	Description
	Namespace	(inherited from Adapter)
	Settings	(inherited from Adapter)

See Also

Reference

[SiebelAdapter Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapter.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

Public Property AcceptCredentialsInUri As Boolean
```

C#

```
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.BatchFields Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property BatchFields As Boolean
```

C#

```
public bool BatchFields { get; set; }
```

C++

```
public:  
virtual property bool BatchFields {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_BatchFields ()  
  
/** @property */  
public final void set_BatchFields (boolean value)
```

JScript

```
public final function get BatchFields () : boolean  
  
public final function set BatchFields (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.EnableConnectionPooling Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

Public Property EnableConnectionPooling As Boolean
```

C#

```
public bool EnableConnectionPooling { get; set; }
```

C++

```
public:
virtual property bool EnableConnectionPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableConnectionPooling ()

/** @property */
public final void set_EnableConnectionPooling (boolean value)
```

JScript

```
public final function get EnableConnectionPooling () : boolean

public final function set EnableConnectionPooling (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.EnablePerformanceCounters Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnablePerformanceCounters As Boolean
```

C#

```
public bool EnablePerformanceCounters { get; set; }
```

C++

```
public:  
virtual property bool EnablePerformanceCounters {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnablePerformanceCounters ()  
  
/** @property */  
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean  
  
public final function set EnablePerformanceCounters (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.IdleConnectionTimeout Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property IdleConnectionTimeout As TimeSpan
```

C#

```
public TimeSpan IdleConnectionTimeout { get; set; }
```

C++

```
public:  
virtual property TimeSpan IdleConnectionTimeout {  
    TimeSpan get () sealed;  
    void set (TimeSpan value) sealed;  
}
```

J#

```
/** @property */  
public final TimeSpan get_IdleConnectionTimeout ()  
  
/** @property */  
public final void set_IdleConnectionTimeout (TimeSpan value)
```

JScript

```
public final function get IdleConnectionTimeout () : TimeSpan  
  
public final function set IdleConnectionTimeout (value : TimeSpan)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.LogData Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property LogData As Boolean
```

C#

```
public bool LogData { get; set; }
```

C++

```
public:  
virtual property bool LogData {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_LogData ()  
  
/** @property */  
public final void set_LogData (boolean value)
```

JScript

```
public final function get LogData () : boolean  
  
public final function set LogData (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.MaxConnectionsPerSystem Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property MaxConnectionsPerSystem As Integer
```

C#

```
public int MaxConnectionsPerSystem { get; set; }
```

C++

```
public:  
virtual property int MaxConnectionsPerSystem {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_MaxConnectionsPerSystem ()  
  
/** @property */  
public final void set_MaxConnectionsPerSystem (int value)
```

JScript

```
public final function get MaxConnectionsPerSystem () : int  
  
public final function set MaxConnectionsPerSystem (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

SiebelAdapter Members
Microsoft.Adapters.Siebel Namespace

SiebelAdapter.Scheme Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Scheme As String
```

C#

```
public override string Scheme { get; }
```

C++

```
public:  
virtual property String^ Scheme {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapter Class](#)

[SiebelAdapter Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Class SiebelAdapterExtensionElement  
    Inherits BindingElementExtensionElement
```

C#

```
public class SiebelAdapterExtensionElement : BindingElementExtensionElement
```

C++

```
public ref class SiebelAdapterExtensionElement : public BindingElementExtensionElement
```

J#

```
public class SiebelAdapterExtensionElement extends BindingElementExtensionElement
```

JScript

```
public class SiebelAdapterExtensionElement extends BindingElementExtensionElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BindingElementExtensionElement](#)

Microsoft.Adapters.Siebel.SiebelAdapterExtensionElement

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement Members

The following tables list the members exposed by the [SiebelAdapterExtensionElement](#) type.

Public Constructors

Name	Description
SiebelAdapterExtensionElement	

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 BatchFields	
 BindingElementType	Overridden.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableConnectionPooling	
 EnablePerformanceCounters	
 IdleConnectionTimeout	
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LogData	
 MaxConnectionsPerSystem	

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	Overridden.

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 ApplyConfiguration	Overridden.
 CopyFrom	Overridden.
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 CreateBindingElement	Overridden.
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 InitializeFrom	Overridden.
 IsModified	(inherited from ServiceModelExtensionElement)
 ListErrors	(inherited from ConfigurationElement)

 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ServiceModelExtensionElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ServiceModelExtensionElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement Constructor

Initializes a new instance of the [SiebelAdapterExtensionElement](#) Class.

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelAdapterExtensionElement ()
```

C++

```
public:  
SiebelAdapterExtensionElement ()
```

J#

```
public SiebelAdapterExtensionElement ()
```

JScript

```
public function SiebelAdapterExtensionElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	Overridden.
	CopyFrom	Overridden.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBindingElement	Overridden.
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	Overridden.
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[SiebelAdapterExtensionElement Class](#)
[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.ApplyConfiguration Method

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

Public Overrides Sub ApplyConfiguration ( _
    bindingElement As BindingElement _
)
```

C#

```
public override void ApplyConfiguration (
    BindingElement bindingElement
)
```

C++

```
public:
virtual void ApplyConfiguration (
    BindingElement^ bindingElement
) override
```

J#

```
public void ApplyConfiguration (
    BindingElement bindingElement
)
```

JScript

```
public override function ApplyConfiguration (
    bindingElement : BindingElement
)
```

Parameters

bindingElement

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.CopyFrom Method

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.CreateBindingElement Method

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Protected Overrides Function CreateBindingElement As BindingElement
```

C#

```
protected override BindingElement CreateBindingElement ()
```

C++

```
protected:  
virtual BindingElement^ CreateBindingElement () override
```

J#

```
protected BindingElement CreateBindingElement ()
```

JScript

```
protected override function CreateBindingElement () : BindingElement
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.InitializeFrom Method

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Protected Overrides Sub InitializeFrom ( _  
    bindingElement As BindingElement _  
)
```

C#

```
protected override void InitializeFrom (  
    BindingElement bindingElement  
)
```

C++

```
protected:  
virtual void InitializeFrom (  
    BindingElement^ bindingElement  
) override
```

J#

```
protected void InitializeFrom (  
    BindingElement bindingElement  
)
```

JScript

```
protected override function InitializeFrom (  
    bindingElement : BindingElement  
)
```

Parameters

bindingElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	AcceptCredentialsInUri	
	BatchFields	
	BindingElementType	Overridden.
	ConfigurationElementName	(inherited from ServiceModelExtensionElement)
	ElementInformation	(inherited from ConfigurationElement)
	EnableConnectionPooling	
	EnablePerformanceCounters	
	IdleConnectionTimeout	
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	LogData	
	MaxConnectionsPerSystem	

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	Overridden.

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("acceptCredentialsInUri", DefaultValue:=False)> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[ConfigurationPropertyAttribute("acceptCredentialsInUri", DefaultValue=false)]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"acceptCredentialsInUri", DefaultValue=false)]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.BatchFields Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("batchFields", DefaultValue:=False)> _
Public Property BatchFields As Boolean
```

C#

```
[ConfigurationPropertyAttribute("batchFields", DefaultValue=false)]
public bool BatchFields { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"batchFields", DefaultValue=false)]
public:
virtual property bool BatchFields {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_BatchFields ()

/** @property */
public final void set_BatchFields (boolean value)
```

JScript

```
public final function get BatchFields () : boolean

public final function set BatchFields (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.BindingElementType Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property BindingElementType As Type
```

C#

```
public override Type BindingElementType { get; }
```

C++

```
public:  
virtual property Type^ BindingElementType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_BindingElementType ()
```

JScript

```
public override function get BindingElementType () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.EnableConnectionPooling Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enableConnectionPooling", DefaultValue:=True)> _
Public Property EnableConnectionPooling As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableConnectionPooling", DefaultValue=true)]
public bool EnableConnectionPooling { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableConnectionPooling", DefaultValue=true)]
public:
virtual property bool EnableConnectionPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableConnectionPooling ()

/** @property */
public final void set_EnableConnectionPooling (boolean value)
```

JScript

```
public final function get EnableConnectionPooling () : boolean

public final function set EnableConnectionPooling (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.EnablePerformanceCounters Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue:=False)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue=false)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enablePerformanceCounters", DefaultValue=false)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.IdleConnectionTimeout Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("idleConnectionTimeout", DefaultValue="00:01:00")> _
Public Property IdleConnectionTimeout As TimeSpan
```

C#

```
[ConfigurationPropertyAttribute("idleConnectionTimeout", DefaultValue="00:01:00")]
public TimeSpan IdleConnectionTimeout { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"idleConnectionTimeout", DefaultValue=L"00:01:00")]
public:
virtual property TimeSpan IdleConnectionTimeout {
    TimeSpan get () sealed;
    void set (TimeSpan value) sealed;
}
```

J#

```
/** @property */
public final TimeSpan get_IdleConnectionTimeout ()

/** @property */
public final void set_IdleConnectionTimeout (TimeSpan value)
```

JScript

```
public final function get IdleConnectionTimeout () : TimeSpan
public final function set IdleConnectionTimeout (value : TimeSpan)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.LogData Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("logData", DefaultValue:=False)> _
Public Property LogData As Boolean
```

C#

```
[ConfigurationPropertyAttribute("logData", DefaultValue=false)]
public bool LogData { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"logData", DefaultValue=false)]
public:
virtual property bool LogData {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_LogData ()

/** @property */
public final void set_LogData (boolean value)
```

JScript

```
public final function get LogData () : boolean

public final function set LogData (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.MaxConnectionsPerSystem Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("maxConnectionsPerSystem", DefaultValue:=20)> _
Public Property MaxConnectionsPerSystem As Integer
```

C#

```
[ConfigurationPropertyAttribute("maxConnectionsPerSystem", DefaultValue=20)]
public int MaxConnectionsPerSystem { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"maxConnectionsPerSystem", DefaultValue=20)]
public:
virtual property int MaxConnectionsPerSystem {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionsPerSystem ()

/** @property */
public final void set_MaxConnectionsPerSystem (int value)
```

JScript

```
public final function get MaxConnectionsPerSystem () : int

public final function set MaxConnectionsPerSystem (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelAdapterExtensionElement.Properties Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Protected Overrides ReadOnly Property Properties As ConfigurationPropertyCollection
```

C#

```
protected override ConfigurationPropertyCollection Properties { get; }
```

C++

```
protected:  
virtual property ConfigurationPropertyCollection^ Properties {  
    ConfigurationPropertyCollection^ get () override;  
}
```

J#

```
/** @property */  
protected ConfigurationPropertyCollection get_Properties ()
```

JScript

```
protected override function get Properties () : ConfigurationPropertyCollection
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelAdapterExtensionElement Class](#)

[SiebelAdapterExtensionElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelBinding  
    Inherits CommonBinding
```

C#

```
public sealed class SiebelBinding : CommonBinding
```

C++

```
public ref class SiebelBinding sealed : public CommonBinding
```

J#

```
public final class SiebelBinding extends CommonBinding
```

JScript

```
public final class SiebelBinding extends CommonBinding
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.Binding](#)

[Microsoft.ServiceModel.Channels.Common.AdapterBinding](#)

[Microsoft.Adapters.AdapterUtilities.CommonBinding](#)

Microsoft.Adapters.Siebel.SiebelBinding

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Members

The following tables list the members exposed by the [SiebelBinding](#) type.

Public Constructors

Name	Description
SiebelBinding	Overloaded.

Top

Public Properties

Name	Description
 AcceptCredentialsInUri	
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 BatchFields	
 CloseTimeout	(inherited from Binding)
 ConnectionUriType	Overridden.
 DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
 EnableBizTalkCompatibilityMode	(inherited from CommonBinding)
 EnableConnectionPooling	
 EnablePerformanceCounters	
 IdleConnectionTimeout	
 LogData	
 MaxConnectionsPerSystem	
 MessageVersion	(inherited from Binding)
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 OpenTimeout	(inherited from Binding)
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden.
 SendTimeout	(inherited from Binding)
 SupportsCertificateAuthentication	(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden.
 SupportsMetadataGet	Overridden.
 SupportsMetadataSearch	Overridden.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 BuildChannelFactory	Overloaded. (inherited from Binding)
 BuildChannelListener	Overloaded. (inherited from Binding)
 CanBuildChannelFactory	Overloaded. (inherited from Binding)
 CanBuildChannelListener	Overloaded. (inherited from Binding)
 CreateBindingElements	Overridden.
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetProperty	(inherited from Binding)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 Finalize	(inherited from Object)

 [MemberwiseClone](#) (inherited from [Object](#))

[Top](#)

[See Also](#)

Reference

[SiebelBinding Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Constructor

Overload List

Name	Description
SiebelBinding ()	
SiebelBinding (String)	

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Constructor ()

Initializes a new instance of the [SiebelBinding](#) Class.

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")>
Public Sub New
```

C#

```
[SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")]
public SiebelBinding ()
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Performance", L"CA1805:DoNotInitializeUnnecessarily")
]
public:
SiebelBinding ()
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily") */
public SiebelBinding ()
```

JScript

```
SuppressMessageAttribute("Microsoft.Performance", "CA1805:DoNotInitializeUnnecessarily")
public function SiebelBinding ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Constructor (String)

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    configName As String _  
)
```

C#

```
public SiebelBinding (  
    string configName  
)
```

C++

```
public:  
SiebelBinding (  
    String^ configName  
)
```

J#

```
public SiebelBinding (  
    String configName  
)
```

JScript

```
public function SiebelBinding (  
    configName : String  
)
```

Parameters

configName

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelBinding Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.CreateBindingElements Method

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Overrides NotOverridable Function CreateBindingElements As BindingElementCollection
```

C#

```
public override sealed BindingElementCollection CreateBindingElements ()
```

C++

```
public:  
virtual BindingElementCollection^ CreateBindingElements () override sealed
```

J#

```
public final BindingElementCollection CreateBindingElements ()
```

JScript

```
public override final function CreateBindingElements () : BindingElementCollection
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding Properties

Public Properties

	Name	Description
	AcceptCredentialsInUri	
	AssuresOrderedDelivery	(inherited from AdapterBinding)
	BatchFields	
	CloseTimeout	(inherited from Binding)
	ConnectionUriType	Overridden.
	DefaultXsdFileNamePrefix	(inherited from AdapterBinding)
	EnableBizTalkCompatibilityMode	(inherited from CommonBinding)
	EnableConnectionPooling	
	EnablePerformanceCounters	
	IdleConnectionTimeout	
	LogData	
	MaxConnectionsPerSystem	
	MessageVersion	(inherited from Binding)
	Name	(inherited from Binding)
	Namespace	(inherited from Binding)
	OpenTimeout	(inherited from Binding)
	QueuedDelivery	(inherited from AdapterBinding)
	ReceiveTimeout	(inherited from Binding)
	Scheme	Overridden.
	SendTimeout	(inherited from Binding)
	SupportsCertificateAuthentication	(inherited from AdapterBinding)
	SupportsMetadataBrowse	Overridden.
	SupportsMetadataGet	Overridden.
	SupportsMetadataSearch	Overridden.
	SupportsUserNameAuthentication	(inherited from AdapterBinding)
	SupportsWindowsAuthentication	(inherited from AdapterBinding)

See Also

Reference

[SiebelBinding Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

Public Property AcceptCredentialsInUri As Boolean
```

C#

```
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.BatchFields Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property BatchFields As Boolean
```

C#

```
public bool BatchFields { get; set; }
```

C++

```
public:  
virtual property bool BatchFields {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_BatchFields ()  
  
/** @property */  
public final void set_BatchFields (boolean value)
```

JScript

```
public final function get BatchFields () : boolean  
  
public final function set BatchFields (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.ConnectionUriType Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property ConnectionUriType As Type
```

C#

```
public override Type ConnectionUriType { get; }
```

C++

```
public:  
virtual property Type^ ConnectionUriType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_ConnectionUriType ()
```

JScript

```
public override function get ConnectionUriType () : Type
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.EnableConnectionPooling Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnableConnectionPooling As Boolean
```

C#

```
public bool EnableConnectionPooling { get; set; }
```

C++

```
public:  
virtual property bool EnableConnectionPooling {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnableConnectionPooling ()  
  
/** @property */  
public final void set_EnableConnectionPooling (boolean value)
```

JScript

```
public final function get EnableConnectionPooling () : boolean  
  
public final function set EnableConnectionPooling (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.EnablePerformanceCounters Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnablePerformanceCounters As Boolean
```

C#

```
public bool EnablePerformanceCounters { get; set; }
```

C++

```
public:  
virtual property bool EnablePerformanceCounters {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnablePerformanceCounters ()  
  
/** @property */  
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean  
  
public final function set EnablePerformanceCounters (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.IdleConnectionTimeout Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property IdleConnectionTimeout As TimeSpan
```

C#

```
public TimeSpan IdleConnectionTimeout { get; set; }
```

C++

```
public:  
virtual property TimeSpan IdleConnectionTimeout {  
    TimeSpan get () sealed;  
    void set (TimeSpan value) sealed;  
}
```

J#

```
/** @property */  
public final TimeSpan get_IdleConnectionTimeout ()  
  
/** @property */  
public final void set_IdleConnectionTimeout (TimeSpan value)
```

JScript

```
public final function get IdleConnectionTimeout () : TimeSpan  
  
public final function set IdleConnectionTimeout (value : TimeSpan)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.LogData Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Property LogData As Boolean
```

C#

```
public bool LogData { get; set; }
```

C++

```
public:  
virtual property bool LogData {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_LogData ()  
  
/** @property */  
public final void set_LogData (boolean value)
```

JScript

```
public final function get LogData () : boolean  
  
public final function set LogData (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.MaxConnectionsPerSystem Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

Public Property MaxConnectionsPerSystem As Integer
```

C#

```
public int MaxConnectionsPerSystem { get; set; }
```

C++

```
public:
virtual property int MaxConnectionsPerSystem {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionsPerSystem ()

/** @property */
public final void set_MaxConnectionsPerSystem (int value)
```

JScript

```
public final function get MaxConnectionsPerSystem () : int

public final function set MaxConnectionsPerSystem (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

SiebelBinding Members
Microsoft.Adapters.Siebel Namespace

SiebelBinding.Scheme Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Scheme As String
```

C#

```
public override string Scheme { get; }
```

C++

```
public:  
virtual property String^ Scheme {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.SupportsMetadataBrowse Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataBrowse As Boolean
```

C#

```
public override bool SupportsMetadataBrowse { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataBrowse {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataBrowse ()
```

JScript

```
public override function get SupportsMetadataBrowse () : boolean
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.SupportsMetadataGet Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataGet As Boolean
```

C#

```
public override bool SupportsMetadataGet { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataGet {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataGet ()
```

JScript

```
public override function get SupportsMetadataGet () : boolean
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBinding.SupportsMetadataSearch Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataSearch As Boolean
```

C#

```
public override bool SupportsMetadataSearch { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataSearch {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataSearch ()
```

JScript

```
public override function get SupportsMetadataSearch () : boolean
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBinding Class](#)

[SiebelBinding Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelBindingConfigElement  
    Inherits CommonStandardBindingElement
```

C#

```
public sealed class SiebelBindingConfigElement : CommonStandardBindingElement
```

C++

```
public ref class SiebelBindingConfigElement sealed : public CommonStandardBindingElement
```

J#

```
public final class SiebelBindingConfigElement extends CommonStandardBindingElement
```

JScript

```
public final class SiebelBindingConfigElement extends CommonStandardBindingElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.StandardBindingElement](#)

[Microsoft.Adapters.AdapterUtilities.CommonStandardBindingElement](#)

Microsoft.Adapters.Siebel.SiebelBindingConfigElement

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Members

The following tables list the members exposed by the [SiebelBindingConfigElement](#) type.

Public Constructors

Name	Description
SiebelBindingConfigElement	Overloaded.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 BatchFields	
 CloseTimeout	(inherited from StandardBindingElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	(inherited from CommonStandardBindingElement)
 EnableConnectionPooling	
 EnablePerformanceCounters	
 IdleConnectionTimeout	
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 LogData	
 MaxConnectionsPerSystem	
 Name	(inherited from StandardBindingElement)
 OpenTimeout	(inherited from StandardBindingElement)
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)

Top

Protected Properties

Name	Description
 BindingElementType	(inherited from StandardBindingElement)
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from CommonStandardBindingElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 ApplyConfiguration	(inherited from StandardBindingElement)
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)

 InitializeFrom	(inherited from CommonStandardBindingElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnApplyConfiguration	(inherited from StandardBindingElement)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from StandardBindingElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[SiebelBindingConfigElement Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Constructor

Overload List

Name	Description
SiebelBindingConfigElement ()	
SiebelBindingConfigElement (String)	

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Constructor ()

Initializes a new instance of the [SiebelBindingConfigElement](#) Class.

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelBindingConfigElement ()
```

C++

```
public:  
SiebelBindingConfigElement ()
```

J#

```
public SiebelBindingConfigElement ()
```

JScript

```
public function SiebelBindingConfigElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Constructor (String)

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    configName As String _  
)
```

C#

```
public SiebelBindingConfigElement (  
    string configName  
)
```

C++

```
public:  
SiebelBindingConfigElement (  
    String^ configName  
)
```

J#

```
public SiebelBindingConfigElement (  
    String configName  
)
```

JScript

```
public function SiebelBindingConfigElement (  
    configName : String  
)
```

Parameters

configName

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

SiebelBindingConfigElement Members
Microsoft.Adapters.Siebel Namespace

SiebelBindingConfigElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from CommonStandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnApplyConfiguration	(inherited from StandardBindingElement)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from StandardBindingElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[SiebelBindingConfigElement Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	AcceptCredentialsInUri	
	BatchFields	
	CloseTimeout	(inherited from StandardBindingElement)
	ElementInformation	(inherited from ConfigurationElement)
	EnableBizTalkCompatibilityMode	(inherited from CommonStandardBindingElement)
	EnableConnectionPooling	
	EnablePerformanceCounters	
	IdleConnectionTimeout	
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	LogData	
	MaxConnectionsPerSystem	
	Name	(inherited from StandardBindingElement)
	OpenTimeout	(inherited from StandardBindingElement)
	ReceiveTimeout	(inherited from StandardBindingElement)
	SendTimeout	(inherited from StandardBindingElement)

Protected Properties

	Name	Description
	BindingElementType	(inherited from StandardBindingElement)
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from CommonStandardBindingElement)

See Also

Reference

[SiebelBindingConfigElement Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("acceptCredentialsInUri")> _
<BrowsableAttribute(False)> _
Public Property AcceptCredentialsInUri As Boolean
```

C#

```
[ConfigurationPropertyAttribute("acceptCredentialsInUri")]
[BrowsableAttribute(false)]
public bool AcceptCredentialsInUri { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"acceptCredentialsInUri")]
[BrowsableAttribute(false)]
public:
virtual property bool AcceptCredentialsInUri {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AcceptCredentialsInUri ()

/** @property */
public final void set_AcceptCredentialsInUri (boolean value)
```

JScript

```
public final function get AcceptCredentialsInUri () : boolean

public final function set AcceptCredentialsInUri (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.BatchFields Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("batchFields")> _
<CategoryAttribute("Performance")> _
Public Property BatchFields As Boolean
```

C#

```
[ConfigurationPropertyAttribute("batchFields")]
[CategoryAttribute("Performance")]
public bool BatchFields { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"batchFields")]
[CategoryAttribute(L"Performance")]
public:
virtual property bool BatchFields {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_BatchFields ()

/** @property */
public final void set_BatchFields (boolean value)
```

JScript

```
public final function get BatchFields () : boolean

public final function set BatchFields (value : boolean)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.EnableConnectionPooling Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enableConnectionPooling")> _
<CategoryAttribute("Connection")> _
Public Property EnableConnectionPooling As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableConnectionPooling")]
[CategoryAttribute("Connection")]
public bool EnableConnectionPooling { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableConnectionPooling")]
[CategoryAttribute(L"Connection")]
public:
virtual property bool EnableConnectionPooling {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableConnectionPooling ()

/** @property */
public final void set_EnableConnectionPooling (boolean value)
```

JScript

```
public final function get EnableConnectionPooling () : boolean

public final function set EnableConnectionPooling (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.EnablePerformanceCounters Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("enablePerformanceCounters")> _
<CategoryAttribute("Diagnostics")> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enablePerformanceCounters")]
[CategoryAttribute("Diagnostics")]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enablePerformanceCounters")]
[CategoryAttribute(L"Diagnostics")]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.IdleConnectionTimeout Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
<ConfigurationPropertyAttribute("idleConnectionTimeout")> _
Public Property IdleConnectionTimeout As TimeSpan
```

C#

```
[CategoryAttribute("Connection")]
[ConfigurationPropertyAttribute("idleConnectionTimeout")]
public TimeSpan IdleConnectionTimeout { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
[ConfigurationPropertyAttribute(L"idleConnectionTimeout")]
public:
virtual property TimeSpan IdleConnectionTimeout {
    TimeSpan get () sealed;
    void set (TimeSpan value) sealed;
}
```

J#

```
/** @property */
public final TimeSpan get_IdleConnectionTimeout ()

/** @property */
public final void set_IdleConnectionTimeout (TimeSpan value)
```

JScript

```
public final function get IdleConnectionTimeout () : TimeSpan

public final function set IdleConnectionTimeout (value : TimeSpan)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.LogData Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Diagnostics")> _
<ConfigurationPropertyAttribute("logData")> _
Public Property LogData As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[ConfigurationPropertyAttribute("logData")]
public bool LogData { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[ConfigurationPropertyAttribute(L"logData")]
public:
virtual property bool LogData {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_LogData ()

/** @property */
public final void set_LogData (boolean value)
```

JScript

```
public final function get LogData () : boolean

public final function set LogData (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelBindingConfigElement.MaxConnectionsPerSystem Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
<ConfigurationPropertyAttribute("maxConnectionsPerSystem")> _
Public Property MaxConnectionsPerSystem As Integer
```

C#

```
[CategoryAttribute("Connection")]
[ConfigurationPropertyAttribute("maxConnectionsPerSystem")]
public int MaxConnectionsPerSystem { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
[ConfigurationPropertyAttribute(L"maxConnectionsPerSystem")]
public:
virtual property int MaxConnectionsPerSystem {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionsPerSystem ()

/** @property */
public final void set_MaxConnectionsPerSystem (int value)
```

JScript

```
public final function get MaxConnectionsPerSystem () : int
public final function set MaxConnectionsPerSystem (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelBindingConfigElement Class](#)

[SiebelBindingConfigElement Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelConnectionUri  
    Inherits ConnectionUri
```

C#

```
public sealed class SiebelConnectionUri : ConnectionUri
```

C++

```
public ref class SiebelConnectionUri sealed : public ConnectionUri
```

J#

```
public final class SiebelConnectionUri extends ConnectionUri
```

JScript

```
public final class SiebelConnectionUri extends ConnectionUri
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.ServiceModel.Channels.Common.ConnectionUri](#)

Microsoft.Adapters.Siebel.SiebelConnectionUri

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri Members

The following tables list the members exposed by the [SiebelConnectionUri](#) type.

Public Constructors

Name	Description
SiebelConnectionUri	

[Top](#)

Public Properties

Name	Description
 Compression	
 Encryption	
 Language	
 SampleUriString	Overridden.
 SecureUriString	Overridden.
 SiebelEnterpriseServer	
 SiebelGateway	
 SiebelObjectManager	
 SiebelRepository	
 SiebelServer	
 Transport	
 Uri	Overridden.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetAttributes	(inherited from ConnectionUri)
 GetClassName	(inherited from ConnectionUri)
 GetComponentName	(inherited from ConnectionUri)
 GetConverter	(inherited from ConnectionUri)
 GetDefaultEvent	(inherited from ConnectionUri)
 GetDefaultProperty	(inherited from ConnectionUri)
 GetEditor	(inherited from ConnectionUri)
 GetEvents	Overloaded. (inherited from ConnectionUri)
 GetHashCode	(inherited from Object)
 GetProperties	Overloaded. (inherited from ConnectionUri)
 GetPropertyOwner	(inherited from ConnectionUri)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from ConnectionUri)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
  GetQueryStringValue	(inherited from ConnectionUri)
  GetUserInfoValue	(inherited from ConnectionUri)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelConnectionUri Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri Constructor

Initializes a new instance of the [SiebelConnectionUri](#) Class.

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelConnectionUri ()
```

C++

```
public:  
SiebelConnectionUri ()
```

J#

```
public SiebelConnectionUri ()
```

JScript

```
public function SiebelConnectionUri ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetAttributes	(inherited from ConnectionUri)
	GetClassName	(inherited from ConnectionUri)
	GetComponentName	(inherited from ConnectionUri)
	GetConverter	(inherited from ConnectionUri)
	GetDefaultEvent	(inherited from ConnectionUri)
	GetDefaultProperty	(inherited from ConnectionUri)
	GetEditor	(inherited from ConnectionUri)
	GetEvents	Overloaded. (inherited from ConnectionUri)
	GetHashCode	(inherited from Object)
	GetProperties	Overloaded. (inherited from ConnectionUri)
	GetPropertyOwner	(inherited from ConnectionUri)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from ConnectionUri)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
 	GetQueryStringValue	(inherited from ConnectionUri)
 	GetUserInfoValue	(inherited from ConnectionUri)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelConnectionUri Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri Properties

Public Properties

	Name	Description
	Compression	
	Encryption	
	Language	
	SampleUriString	Overridden.
	SecureUriString	Overridden.
	SiebelEnterpriseServer	
	SiebelGateway	
	SiebelObjectManager	
	SiebelRepository	
	SiebelServer	
	Transport	
	Uri	Overridden.

See Also

Reference

[SiebelConnectionUri Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.Compression Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Compression to use (optional)")> _
<CategoryAttribute("Siebel Configuration")> _
<BrowsableAttribute(True)> _
Public Property Compression As String
```

C#

```
[DescriptionAttribute("Compression to use (optional)")]
[CategoryAttribute("Siebel Configuration")]
[BrowsableAttribute(true)]
public string Compression { get; set; }
```

C++

```
[DescriptionAttribute(L"Compression to use (optional)")]
[CategoryAttribute(L"Siebel Configuration")]
[BrowsableAttribute(true)]
public:
property String^ Compression {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Compression ()

/** @property */
public void set_Compression (String value)
```

JScript

```
public function get Compression () : String

public function set Compression (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.Encryption Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Siebel Configuration")> _
<DescriptionAttribute("Encryption to use (optional)")> _
Public Property Encryption As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Siebel Configuration")]
[DescriptionAttribute("Encryption to use (optional)")]
public string Encryption { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Siebel Configuration")]
[DescriptionAttribute(L"Encryption to use (optional)")]
public:
property String^ Encryption {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Encryption ()

/** @property */
public void set_Encryption (String value)
```

JScript

```
public function get Encryption () : String

public function set Encryption (value : String)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.Language Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Language to use (for e.g. enu), optional")> _
<CategoryAttribute("Siebel Configuration")> _
<BrowsableAttribute(True)> _
Public Property Language As String
```

C#

```
[DescriptionAttribute("Language to use (for e.g. enu), optional")]
[CategoryAttribute("Siebel Configuration")]
[BrowsableAttribute(true)]
public string Language { get; set; }
```

C++

```
[DescriptionAttribute(L"Language to use (for e.g. enu), optional")]
[CategoryAttribute(L"Siebel Configuration")]
[BrowsableAttribute(true)]
public:
property String^ Language {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Language ()

/** @property */
public void set_Language (String value)
```

JScript

```
public function get Language () : String

public function set Language (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SampleUriString Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SampleUriString As String
```

C#

```
public override string SampleUriString { get; }
```

C++

```
public:  
virtual property String^ SampleUriString {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_SampleUriString ()
```

JScript

```
public override function get SampleUriString () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SecureUriString Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
Public Overrides ReadOnly Property SecureUriString As String
```

C#

```
[BrowsableAttribute(false)]
public override string SecureUriString { get; }
```

C++

```
[BrowsableAttribute(false)]
public:
virtual property String^ SecureUriString {
String^ get () override;
}
```

J#

```
/** @property */
public String get_SecureUriString ()
```

JScript

```
public override function get SecureUriString () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SiebelEnterpriseServer Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Siebel Configuration")> _
<DescriptionAttribute("Siebel Enterprise Server")> _
<BrowsableAttribute(True)> _
Public Property SiebelEnterpriseServer As String
```

C#

```
[CategoryAttribute("Siebel Configuration")]
[DescriptionAttribute("Siebel Enterprise Server")]
[BrowsableAttribute(true)]
public string SiebelEnterpriseServer { get; set; }
```

C++

```
[CategoryAttribute(L"Siebel Configuration")]
[DescriptionAttribute(L"Siebel Enterprise Server")]
[BrowsableAttribute(true)]
public:
property String^ SiebelEnterpriseServer {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelEnterpriseServer ()

/** @property */
public void set_SiebelEnterpriseServer (String value)
```

JScript

```
public function get SiebelEnterpriseServer () : String

public function set SiebelEnterpriseServer (value : String)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SiebelGateway Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Machine name and port number (For e.g. SiebelServer:2321)")> _
<CategoryAttribute("Siebel Configuration")> _
<BrowsableAttribute(True)> _
Public Property SiebelGateway As String
```

C#

```
[DescriptionAttribute("Machine name and port number (For e.g. SiebelServer:2321)")]
[CategoryAttribute("Siebel Configuration")]
[BrowsableAttribute(true)]
public string SiebelGateway { get; set; }
```

C++

```
[DescriptionAttribute(L"Machine name and port number (For e.g. SiebelServer:2321)")]
[CategoryAttribute(L"Siebel Configuration")]
[BrowsableAttribute(true)]
public:
property String^ SiebelGateway {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelGateway ()

/** @property */
public void set_SiebelGateway (String value)
```

JScript

```
public function get SiebelGateway () : String

public function set SiebelGateway (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SiebelObjectManager Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Siebel object manager")> _
<CategoryAttribute("Siebel Configuration")> _
<BrowsableAttribute(True)> _
Public Property SiebelObjectManager As String
```

C#

```
[DescriptionAttribute("Siebel object manager")]
[CategoryAttribute("Siebel Configuration")]
[BrowsableAttribute(true)]
public string SiebelObjectManager { get; set; }
```

C++

```
[DescriptionAttribute(L"Siebel object manager")]
[CategoryAttribute(L"Siebel Configuration")]
[BrowsableAttribute(true)]
public:
property String^ SiebelObjectManager {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelObjectManager ()

/** @property */
public void set_SiebelObjectManager (String value)
```

JScript

```
public function get SiebelObjectManager () : String

public function set SiebelObjectManager (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SiebelRepository Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Siebel Configuration")> _
<BrowsableAttribute(True)> _
<DescriptionAttribute("Repository to use (required if more than one repository configured o
n Siebel server, optional otherwise)")> _
Public Property SiebelRepository As String
```

C#

```
[CategoryAttribute("Siebel Configuration")]
[BrowsableAttribute(true)]
[DescriptionAttribute("Repository to use (required if more than one repository configured o
n Siebel server, optional otherwise)")]
public string SiebelRepository { get; set; }
```

C++

```
[CategoryAttribute(L"Siebel Configuration")]
[BrowsableAttribute(true)]
[DescriptionAttribute(L"Repository to use (required if more than one repository configured
on Siebel server, optional otherwise)")]
public:
property String^ SiebelRepository {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelRepository ()

/** @property */
public void set_SiebelRepository (String value)
```

JScript

```
public function get SiebelRepository () : String
public function set SiebelRepository (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.SiebelServer Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<DescriptionAttribute("Siebel server (required for siebel 7.5.x versions, optional for other
rs)")> _
<CategoryAttribute("Siebel Configuration")> _
Public Property SiebelServer As String
```

C#

```
[BrowsableAttribute(true)]
[DescriptionAttribute("Siebel server (required for siebel 7.5.x versions, optional for other
rs)")]
[CategoryAttribute("Siebel Configuration")]
public string SiebelServer { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[DescriptionAttribute(L"Siebel server (required for siebel 7.5.x versions, optional for other
rs)")]
[CategoryAttribute(L"Siebel Configuration")]
public:
property String^ SiebelServer {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelServer ()

/** @property */
public void set_SiebelServer (String value)
```

JScript

```
public function get SiebelServer () : String

public function set SiebelServer (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.Transport Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(True)> _
<CategoryAttribute("Siebel Configuration")> _
<DescriptionAttribute("Transport to use (optional)")> _
Public Property Transport As String
```

C#

```
[BrowsableAttribute(true)]
[CategoryAttribute("Siebel Configuration")]
[DescriptionAttribute("Transport to use (optional)")]
public string Transport { get; set; }
```

C++

```
[BrowsableAttribute(true)]
[CategoryAttribute(L"Siebel Configuration")]
[DescriptionAttribute(L"Transport to use (optional)")]
public:
property String^ Transport {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Transport ()

/** @property */
public void set_Transport (String value)
```

JScript

```
public function get Transport () : String

public function set Transport (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConnectionUri.Uri Property

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
Public Overrides Property Uri As Uri
```

C#

```
[BrowsableAttribute(false)]
public override Uri Uri { get; set; }
```

C++

```
[BrowsableAttribute(false)]
public:
virtual property Uri^ Uri {
    Uri^ get () override;
    void set (Uri^ value) override;
}
```

J#

```
/** @property */
public Uri get Uri ()

/** @property */
public void set Uri (Uri value)
```

JScript

```
public override function get Uri () : Uri

public override function set Uri (value : Uri)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionUri Class](#)

[SiebelConnectionUri Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants Class

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelConstants
```

C#

```
public static class SiebelConstants
```

C++

```
public ref class SiebelConstants abstract sealed
```

J#

```
public final class SiebelConstants
```

JScript

```
public final class SiebelConstants
```

Inheritance Hierarchy [System.Object](#)

Microsoft.Adapters.Siebel.SiebelConstants

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants Members

The following tables list the members exposed by the [SiebelConstants](#) type.

Public Fields

	Name	Description
	PerfCategory	
	PerfCounterLobLatency	

Top

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Top

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

Top

See Also

Reference

[SiebelConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants Fields

Public Fields

	Name	Description
 	PerfCategory	
 	PerfCounterLobLatency	

See Also

Reference

[SiebelConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants.PerfCategory Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId:="Perf")> _
Public Const PerfCategory As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")]
public const string PerfCategory
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId=L"Perf")]
public:
literal String^ PerfCategory
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly", MessageId="Perf") */
public static final String PerfCategory
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")
public const var PerfCategory : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConstants Class](#)

[SiebelConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants.PerfCounterLobLatency Field

Namespace: Microsoft.Adapters.Siebel

Assembly: Microsoft.Adapters.Siebel (in microsoft.adapters.siebel.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId:="Perf")> _
Public Const PerfCounterLobLatency As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")]
public const string PerfCounterLobLatency
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Naming", L"CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId=L"Perf")]
public:
    literal String^ PerfCounterLobLatency
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly", MessageId="Perf") */
public static final String PerfCounterLobLatency
```

JScript

```
SuppressMessageAttribute("Microsoft.Naming", "CA1704:IdentifiersShouldBeSpelledCorrectly",
MessageId="Perf")
public const var PerfCounterLobLatency : String
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConstants Class](#)

[SiebelConstants Members](#)

[Microsoft.Adapters.Siebel Namespace](#)

SiebelConstants Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelConstants Class](#)

[Microsoft.Adapters.Siebel Namespace](#)

Microsoft.Data.SiebelClient Namespace

Classes

Class	Description
SiebelClientFactory	
SiebelCommand	
SiebelCommandException	
SiebelConnection	
SiebelConnectionException	
SiebelConnectionStringBuilder	
SiebelDataReaderException	
SiebelException	
SiebelParameter	
SiebelParameterException	
SiebelParserException	

SiebelClientFactory Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Class SiebelClientFactory  
    Inherits DbProviderFactory
```

C#

```
public class SiebelClientFactory : DbProviderFactory
```

C++

```
public ref class SiebelClientFactory : public DbProviderFactory
```

J#

```
public class SiebelClientFactory extends DbProviderFactory
```

JScript

```
public class SiebelClientFactory extends DbProviderFactory
```

Inheritance Hierarchy [System.Object](#)

[System.Data.Common.DbProviderFactory](#)

Microsoft.Data.SiebelClient.SiebelClientFactory

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory Members

The following tables list the members exposed by the [SiebelClientFactory](#) type.

Public Fields

	Name	Description
	Instance	

[Top](#)

Public Properties

	Name	Description
	CanCreateDataSourceEnumerator	(inherited from DbProviderFactory)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	CreateCommand	Overridden.
	CreateCommandBuilder	Overridden.
	CreateConnection	Overridden.
	CreateConnectionStringBuilder	Overridden.
	CreateDataAdapter	Overridden.
	CreateDataSourceEnumerator	Overridden.
	CreateParameter	Overridden.
	CreatePermission	Overridden.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelClientFactory Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory Fields

Public Fields

	Name	Description
	Instance	

See Also

Reference

[SiebelClientFactory Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.Instance Field

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Security", "CA2104:DoNotDeclareReadOnlyMutableReferenceTypes")> _
Public Shared ReadOnly Instance As SiebelClientFactory
```

C#

```
[SuppressMessageAttribute("Microsoft.Security", "CA2104:DoNotDeclareReadOnlyMutableReferenceTypes")]
public static readonly SiebelClientFactory Instance
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Security", L"CA2104:DoNotDeclareReadOnlyMutableReferenceTypes")]
public:
static initonly SiebelClientFactory^ Instance
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Security", "CA2104:DoNotDeclareReadOnlyMutableReferenceTypes") */
public static final SiebelClientFactory Instance
```

JScript

```
SuppressMessageAttribute("Microsoft.Security", "CA2104:DoNotDeclareReadOnlyMutableReferenceTypes")
public static final var Instance : SiebelClientFactory
```

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	CreateCommand	Overridden.
	CreateCommandBuilder	Overridden.
	CreateConnection	Overridden.
	CreateConnectionStringBuilder	Overridden.
	CreateDataAdapter	Overridden.
	CreateDataSourceEnumerator	Overridden.
	CreateParameter	Overridden.
	CreatePermission	Overridden.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelClientFactory Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateCommand Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateCommand As DbCommand
```

C#

```
public override DbCommand CreateCommand ()
```

C++

```
public:  
virtual DbCommand^ CreateCommand () override
```

J#

```
public DbCommand CreateCommand ()
```

JScript

```
public override function CreateCommand () : DbCommand
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateCommandBuilder Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateCommandBuilder As DbCommandBuilder
```

C#

```
public override DbCommandBuilder CreateCommandBuilder ()
```

C++

```
public:  
virtual DbCommandBuilder^ CreateCommandBuilder () override
```

J#

```
public DbCommandBuilder CreateCommandBuilder ()
```

JScript

```
public override function CreateCommandBuilder () : DbCommandBuilder
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateConnection Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Function CreateConnection As DbConnection
```

C#

```
public override DbConnection CreateConnection ()
```

C++

```
public:  
virtual DbConnection^ CreateConnection () override
```

J#

```
public DbConnection CreateConnection ()
```

JScript

```
public override function CreateConnection () : DbConnection
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateConnectionStringBuilder Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateConnectionStringBuilder As DbConnectionStringBuilder
```

C#

```
public override DbConnectionStringBuilder CreateConnectionStringBuilder ()
```

C++

```
public:  
virtual DbConnectionStringBuilder^ CreateConnectionStringBuilder () override
```

J#

```
public DbConnectionStringBuilder CreateConnectionStringBuilder ()
```

JScript

```
public override function CreateConnectionStringBuilder () : DbConnectionStringBuilder
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateDataAdapter Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Function CreateDataAdapter As DbDataAdapter
```

C#

```
public override DbDataAdapter CreateDataAdapter ()
```

C++

```
public:  
virtual DbDataAdapter^ CreateDataAdapter () override
```

J#

```
public DbDataAdapter CreateDataAdapter ()
```

JScript

```
public override function CreateDataAdapter () : DbDataAdapter
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateDataSourceEnumerator Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateDataSourceEnumerator As DbDataSourceEnumerator
```

C#

```
public override DbDataSourceEnumerator CreateDataSourceEnumerator ()
```

C++

```
public:  
virtual DbDataSourceEnumerator^ CreateDataSourceEnumerator () override
```

J#

```
public DbDataSourceEnumerator CreateDataSourceEnumerator ()
```

JScript

```
public override function CreateDataSourceEnumerator () : DbDataSourceEnumerator
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreateParameter Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function CreateParameter As DbParameter
```

C#

```
public override DbParameter CreateParameter ()
```

C++

```
public:  
virtual DbParameter^ CreateParameter () override
```

J#

```
public DbParameter CreateParameter ()
```

JScript

```
public override function CreateParameter () : DbParameter
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory.CreatePermission Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Function CreatePermission ( _
    state As PermissionState _
) As CodeAccessPermission
```

C#

```
public override CodeAccessPermission CreatePermission (
    PermissionState state
)
```

C++

```
public:
virtual CodeAccessPermission^ CreatePermission (
    PermissionState state
) override
```

J#

```
public CodeAccessPermission CreatePermission (
    PermissionState state
)
```

JScript

```
public override function CreatePermission (
    state : PermissionState
) : CodeAccessPermission
```

Parameters

state

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelClientFactory Class](#)

[SiebelClientFactory Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelClientFactory Properties

Public Properties

	Name	Description
	CanCreateDataSourceEnumerator	(inherited from DbProviderFactory)

See Also

Reference

[SiebelClientFactory Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelCommand  
    Inherits DbCommand
```

C#

```
public sealed class SiebelCommand : DbCommand
```

C++

```
public ref class SiebelCommand sealed : public DbCommand
```

J#

```
public final class SiebelCommand extends DbCommand
```

JScript

```
public final class SiebelCommand extends DbCommand
```

Inheritance Hierarchy [System.Object](#)

[System.MarshalByRefObject](#)

[System.ComponentModel.Component](#)

[System.Data.Common.DbCommand](#)

Microsoft.Data.SiebelClient.SiebelCommand

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Members

The following tables list the members exposed by the [SiebelCommand](#) type.

Public Constructors

Name	Description
SiebelCommand	Overloaded.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 CommandText	Overridden.
 CommandTimeout	Overridden.
 CommandType	Overridden.
 Connection	(inherited from DbCommand)
 Container	(inherited from Component)
 DesignTimeVisible	Overridden.
 Parameters	(inherited from DbCommand)
 Site	(inherited from Component)
 Transaction	(inherited from DbCommand)
 UpdatedRowSource	Overridden.

Top

Protected Properties

Name	Description
 CanRaiseEvents	(inherited from Component)
 DbConnection	(inherited from DbCommand)
 DbParameterCollection	(inherited from DbCommand)
 DbTransaction	(inherited from DbCommand)
 DesignMode	(inherited from Component)
 Events	(inherited from Component)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 Cancel	Overridden.
 CreateObjRef	(inherited from MarshalByRefObject)
 CreateParameter	(inherited from DbCommand)
 Dispose	Overloaded. (inherited from Component)
 Equals	Overloaded. (inherited from Object)
 ExecuteNonQuery	Overridden.
 ExecuteReader	Overloaded. (inherited from DbCommand)
 ExecuteScalar	Overridden.
 GetHashCode	(inherited from Object)
 GetLifetimeService	(inherited from MarshalByRefObject)
 GetType	(inherited from Object)
 InitializeLifetimeService	(inherited from MarshalByRefObject)
 Prepare	Overridden.
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Component)

Top

Protected Methods

Name	Description
 CreateDbParameter	(inherited from DbCommand)
 Dispose	Overloaded. (inherited from Component)
 ExecuteDbDataReader	(inherited from DbCommand)
 Finalize	(inherited from Component)

 GetService	(inherited from Component)
 MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)

[Top](#)

Public Events

	Name	Description
	Disposed	(inherited from Component)

[Top](#)

Explicit Interface Implementations

Name	Description
System.Data.IDbCommand.CreateParameter	(inherited from DbCommand)
System.Data.IDbCommand.ExecuteReader	Overloaded. (inherited from DbCommand)
System.Data.IDbCommand.Connection	(inherited from DbCommand)
System.Data.IDbCommand.Parameters	(inherited from DbCommand)
System.Data.IDbCommand.Transaction	(inherited from DbCommand)

[Top](#)

See Also

Reference

[SiebelCommand Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Constructor

Overload List

Name	Description
SiebelCommand ()	
SiebelCommand (SiebelConnection)	

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Constructor ()

Initializes a new instance of the [SiebelCommand](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelCommand ()
```

C++

```
public:  
SiebelCommand ()
```

J#

```
public SiebelCommand ()
```

JScript

```
public function SiebelCommand ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Constructor (SiebelConnection)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    connection As SiebelConnection _  
)
```

C#

```
public SiebelCommand (  
    SiebelConnection connection  
)
```

C++

```
public:  
SiebelCommand (  
    SiebelConnection^ connection  
)
```

J#

```
public SiebelCommand (  
    SiebelConnection connection  
)
```

JScript

```
public function SiebelCommand (  
    connection : SiebelConnection  
)
```

Parameters

connection

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Cancel	Overridden.
	CreateObjRef	(inherited from MarshalByRefObject)
	CreateParameter	(inherited from DbCommand)
	Dispose	Overloaded. (inherited from Component)
	Equals	Overloaded. (inherited from Object)
	ExecuteNonQuery	Overridden.
	ExecuteReader	Overloaded. (inherited from DbCommand)
	ExecuteScalar	Overridden.
	GetHashCode	(inherited from Object)
	GetLifetimeService	(inherited from MarshalByRefObject)
	GetType	(inherited from Object)
	InitializeLifetimeService	(inherited from MarshalByRefObject)
	Prepare	Overridden.
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Component)

Protected Methods

	Name	Description
	CreateDbParameter	(inherited from DbCommand)
	Dispose	Overloaded. (inherited from Component)
	ExecuteDbDataReader	(inherited from DbCommand)
	Finalize	(inherited from Component)
	GetService	(inherited from Component)
	MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)

Explicit Interface Implementations

Name	Description
System.Data.IDbCommand.CreateParameter	(inherited from DbCommand)
System.Data.IDbCommand.ExecuteReader	Overloaded. (inherited from DbCommand)

See Also

Reference

[SiebelCommand Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand.Cancel Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Sub Cancel
```

C#

```
public override void Cancel ()
```

C++

```
public:  
virtual void Cancel () override
```

J#

```
public void Cancel ()
```

JScript

```
public override function Cancel ()
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand.ExecuteNonQuery Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Function ExecuteNonQuery As Integer
```

C#

```
public override int ExecuteNonQuery ()
```

C++

```
public:  
virtual int ExecuteNonQuery () override
```

J#

```
public int ExecuteNonQuery ()
```

JScript

```
public override function ExecuteNonQuery () : int
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand.ExecuteScalar Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function ExecuteScalar As Object
```

C#

```
public override Object ExecuteScalar ()
```

C++

```
public:  
virtual Object^ ExecuteScalar () override
```

J#

```
public Object ExecuteScalar ()
```

JScript

```
public override function ExecuteScalar () : Object
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand.Prepare Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Sub Prepare
```

C#

```
public override void Prepare ()
```

C++

```
public:  
virtual void Prepare () override
```

J#

```
public void Prepare ()
```

JScript

```
public override function Prepare ()
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

[SiebelCommand Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	CommandText	Overridden.
	CommandTimeout	Overridden.
	CommandType	Overridden.
	Connection	(inherited from DbCommand)
	Container	(inherited from Component)
	DesignTimeVisible	Overridden.
	Parameters	(inherited from DbCommand)
	Site	(inherited from Component)
	Transaction	(inherited from DbCommand)
	UpdatedRowSource	Overridden.

Protected Properties

	Name	Description
	CanRaiseEvents	(inherited from Component)
	DbConnection	(inherited from DbCommand)
	DbParameterCollection	(inherited from DbCommand)
	DbTransaction	(inherited from DbCommand)
	DesignMode	(inherited from Component)
	Events	(inherited from Component)

Explicit Interface Implementations

	Name	Description
	System.Data.DbCommand.Connection	(inherited from DbCommand)
	System.Data.IDbCommand.Parameters	(inherited from DbCommand)
	System.Data.IDbCommand.Transaction	(inherited from DbCommand)

See Also

Reference

[SiebelCommand Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommand.CommandText Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property CommandText As String
```

C#

```
public override string CommandText { get; set; }
```

C++

```
public:
virtual property String^ CommandText {
String^ get () override;
void set (String^ value) override;
}
```

J#

```
/** @property */
public String get_CommandText ()

/** @property */
public void set_CommandText (String value)
```

JScript

```
public override function get CommandText () : String

public override function set CommandText (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand.CommandTimeout Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property CommandTimeout As Integer
```

C#

```
public override int CommandTimeout { get; set; }
```

C++

```
public:
virtual property int CommandTimeout {
    int get () override;
    void set (int value) override;
}
```

J#

```
/** @property */
public int get_CommandTimeout ()

/** @property */
public void set_CommandTimeout (int value)
```

JScript

```
public override function get CommandTimeout () : int

public override function set CommandTimeout (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand.CommandType Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property CommandType As CommandType
```

C#

```
public override CommandType CommandType { get; set; }
```

C++

```
public:
virtual property CommandType CommandType {
    CommandType get () override;
    void set (CommandType value) override;
}
```

J#

```
/** @property */
public CommandType get_CommandType ()

/** @property */
public void set_CommandType (CommandType value)
```

JScript

```
public override function get CommandType () : CommandType

public override function set CommandType (value : CommandType)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand.DesignTimeVisible Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property DesignTimeVisible As Boolean
```

C#

```
public override bool DesignTimeVisible { get; set; }
```

C++

```
public:  
virtual property bool DesignTimeVisible {  
    bool get () override;  
    void set (bool value) override;  
}
```

J#

```
/** @property */  
public boolean get_DesignTimeVisible ()  
  
/** @property */  
public void set_DesignTimeVisible (boolean value)
```

JScript

```
public override function get DesignTimeVisible () : boolean  
  
public override function set DesignTimeVisible (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand.UpdatedRowSource Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property UpdatedRowSource As UpdateRowSource
```

C#

```
public override UpdateRowSource UpdatedRowSource { get; set; }
```

C++

```
public:
virtual property UpdateRowSource UpdatedRowSource {
    UpdateRowSource get () override;
    void set (UpdateRowSource value) override;
}
```

J#

```
/** @property */
public UpdateRowSource get_UpdatedRowSource ()

/** @property */
public void set_UpdatedRowSource (UpdateRowSource value)
```

JScript

```
public override function get UpdatedRowSource () : UpdateRowSource

public override function set UpdatedRowSource (value : UpdateRowSource)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommand Class](#)

SiebelCommand Members
Microsoft.Data.SiebelClient Namespace

SiebelCommand Events

Public Events

	Name	Description
	Disposed	(inherited from Component)

See Also

Reference

[SiebelCommand Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelCommandException
    Inherits SiebelException
```

C#

```
[SerializableAttribute]
public class SiebelCommandException : SiebelException
```

C++

```
[SerializableAttribute]
public ref class SiebelCommandException : public SiebelException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelCommandException extends SiebelException
```

JScript

```
SerializableAttribute
public class SiebelCommandException extends SiebelException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

[Microsoft.Data.SiebelClient.SiebelException](#)

Microsoft.Data.SiebelClient.SiebelCommandException

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommandException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Members

The following tables list the members exposed by the [SiebelCommandException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelCommandException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelCommandException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HResult	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelCommandException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Constructor

Overload List

Name	Description
SiebelCommandException ()	
SiebelCommandException (String)	
SiebelCommandException (SerializationInfo, StreamingContext)	
SiebelCommandException (String, Exception)	

See Also

Reference

[SiebelCommandException Class](#)

[SiebelCommandException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Constructor ()

Initializes a new instance of the [SiebelCommandException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelCommandException ()
```

C++

```
public:  
SiebelCommandException ()
```

J#

```
public SiebelCommandException ()
```

JScript

```
public function SiebelCommandException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommandException Class](#)

[SiebelCommandException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelCommandException (  
    string message  
)
```

C++

```
public:  
SiebelCommandException (  
    String^ message  
)
```

J#

```
public SiebelCommandException (  
    String message  
)
```

JScript

```
public function SiebelCommandException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommandException Class](#)

SiebelCommandException Members
Microsoft.Data.SiebelClient Namespace

SiebelCommandException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelCommandException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelCommandException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelCommandException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelCommandException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommandException Class](#)

[SiebelCommandException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public SiebelCommandException (  
    string message,  
    Exception inner  
)
```

C++

```
public:  
SiebelCommandException (  
    String^ message,  
    Exception^ inner  
)
```

J#

```
public SiebelCommandException (  
    String message,  
    Exception inner  
)
```

JScript

```
public function SiebelCommandException (  
    message : String,  
    inner : Exception  
)
```

Parameters

message

inner

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelCommandException Class](#)

[SiebelCommandException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
 	Finalize	(inherited from Object)
 	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelCommandException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelCommandException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelCommandException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelConnection  
    Inherits DbConnection
```

C#

```
public sealed class SiebelConnection : DbConnection
```

C++

```
public ref class SiebelConnection sealed : public DbConnection
```

J#

```
public final class SiebelConnection extends DbConnection
```

JScript

```
public final class SiebelConnection extends DbConnection
```

Inheritance Hierarchy [System.Object](#)

[System.MarshalByRefObject](#)

[System.ComponentModel.Component](#)

[System.Data.Common.DbConnection](#)

Microsoft.Data.SiebelClient.SiebelConnection

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Members

The following tables list the members exposed by the [SiebelConnection](#) type.

Public Constructors

Name	Description
SiebelConnection	

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 ConnectionString	Overridden.
 ConnectionTimeout	(inherited from DbConnection)
 Container	(inherited from Component)
 Database	Overridden.
 DataSource	Overridden.
 ServerVersion	Overridden.
 Site	(inherited from Component)
 State	Overridden.

Top

Protected Properties

Name	Description
 CanRaiseEvents	(inherited from Component)
 DbProviderFactory	(inherited from DbConnection)
 DesignMode	(inherited from Component)
 Events	(inherited from Component)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 BeginTransaction	Overloaded. (inherited from DbConnection)
 ChangeDatabase	Overridden.
 Close	Overridden.
 CreateCommand	
 CreateObjRef	(inherited from MarshalByRefObject)
 Dispose	Overloaded. (inherited from Component)
 EnlistTransaction	(inherited from DbConnection)
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetLifetimeService	(inherited from MarshalByRefObject)
 GetSchema	Overloaded. Overridden.
 GetType	(inherited from Object)
 InitializeLifetimeService	(inherited from MarshalByRefObject)
 Open	Overridden.
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Component)

Top

Protected Methods

Name	Description
 BeginDbTransaction	(inherited from DbConnection)
 CreateDbCommand	(inherited from DbConnection)
 Dispose	Overloaded. (inherited from Component)
 Finalize	(inherited from Component)
 GetService	(inherited from Component)
 MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)
 OnStateChange	(inherited from DbConnection)

[Top](#)

Public Events

	Name	Description
	Disposed	(inherited from Component)
	StateChange	(inherited from DbConnection)

[Top](#)

Explicit Interface Implementations

Name	Description
System.Data.IDbConnection.BeginTransaction	Overloaded. (inherited from DbConnection)
System.Data.IDbConnection.CreateCommand	(inherited from DbConnection)

[Top](#)

See Also

Reference

[SiebelConnection Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Constructor

Initializes a new instance of the [SiebelConnection](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelConnection ()
```

C++

```
public:  
SiebelConnection ()
```

J#

```
public SiebelConnection ()
```

JScript

```
public function SiebelConnection ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BeginTransaction	Overloaded. (inherited from DbConnection)
	ChangeDatabase	Overridden.
	Close	Overridden.
	CreateCommand	
	CreateObjRef	(inherited from MarshalByRefObject)
	Dispose	Overloaded. (inherited from Component)
	EnlistTransaction	(inherited from DbConnection)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetLifetimeService	(inherited from MarshalByRefObject)
	GetSchema	Overloaded. Overridden.
	GetType	(inherited from Object)
	InitializeLifetimeService	(inherited from MarshalByRefObject)
	Open	Overridden.
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Component)

Protected Methods

	Name	Description
	BeginDbTransaction	(inherited from DbConnection)
	CreateDbCommand	(inherited from DbConnection)
	Dispose	Overloaded. (inherited from Component)
	Finalize	(inherited from Component)
	GetService	(inherited from Component)
	MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)
	OnStateChange	(inherited from DbConnection)

Explicit Interface Implementations

Name	Description
System.Data.IDbConnection.BeginTransaction	Overloaded. (inherited from DbConnection)
System.Data.IDbConnection.CreateCommand	(inherited from DbConnection)

See Also

Reference

[SiebelConnection Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.ChangeDatabase Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Sub ChangeDatabase ( _
    databaseName As String _
)
```

C#

```
public override void ChangeDatabase (
    string databaseName
)
```

C++

```
public:
virtual void ChangeDatabase (
    String^ databaseName
) override
```

J#

```
public void ChangeDatabase (
    String databaseName
)
```

JScript

```
public override function ChangeDatabase (
    databaseName : String
)
```

Parameters

databaseName

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.Close Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Sub Close
```

C#

```
public override void Close ()
```

C++

```
public:  
virtual void Close () override
```

J#

```
public void Close ()
```

JScript

```
public override function Close ()
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.CreateCommand Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Function CreateCommand As SiebelCommand
```

C#

```
public SiebelCommand CreateCommand ()
```

C++

```
public:  
SiebelCommand^ CreateCommand ()
```

J#

```
public SiebelCommand CreateCommand ()
```

JScript

```
public function CreateCommand () : SiebelCommand
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.GetSchema Method

Overload List

Name	Description
SiebelConnection.GetSchema ()	
SiebelConnection.GetSchema (String)	
SiebelConnection.GetSchema (String, String[])	

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.GetSchema Method ()

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Function GetSchema As DataTable
```

C#

```
public override DataTable GetSchema ()
```

C++

```
public:  
virtual DataTable^ GetSchema () override
```

J#

```
public DataTable GetSchema ()
```

JScript

```
public override function GetSchema () : DataTable
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.GetSchema Method (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Function GetSchema ( _  
    collectionName As String _  
) As DataTable
```

C#

```
public override DataTable GetSchema (  
    string collectionName  
)
```

C++

```
public:  
virtual DataTable^ GetSchema (  
    String^ collectionName  
) override
```

J#

```
public DataTable GetSchema (  
    String collectionName  
)
```

JScript

```
public override function GetSchema (  
    collectionName : String  
) : DataTable
```

Parameters

collectionName

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.GetSchema Method (String, String[])

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Function GetSchema ( _
    collectionName As String, _
    restrictionValues As String() _
) As DataTable
```

C#

```
public override DataTable GetSchema (
    string collectionName,
    string[] restrictionValues
)
```

C++

```
public:
virtual DataTable^ GetSchema (
    String^ collectionName,
    array<String^>^ restrictionValues
) override
```

J#

```
public DataTable GetSchema (
    String collectionName,
    String[] restrictionValues
)
```

JScript

```
public override function GetSchema (
    collectionName : String,
    restrictionValues : String[]
) : DataTable
```

Parameters

collectionName

restrictionValues

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.Open Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Sub Open
```

C#

```
public override void Open ()
```

C++

```
public:  
virtual void Open () override
```

J#

```
public void Open ()
```

JScript

```
public override function Open ()
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	ConnectionString	Overridden.
	ConnectionTimeout	(inherited from DbConnection)
	Container	(inherited from Component)
	Database	Overridden.
	DataSource	Overridden.
	ServerVersion	Overridden.
	Site	(inherited from Component)
	State	Overridden.

Protected Properties

	Name	Description
	CanRaiseEvents	(inherited from Component)
	DbProviderFactory	(inherited from DbConnection)
	DesignMode	(inherited from Component)
	Events	(inherited from Component)

See Also

Reference

[SiebelConnection Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.ConnectionString Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Maintainability", "CA1502:AvoidExcessiveComplexity")>
Public Overrides Property ConnectionString As String
```

C#

```
[SuppressMessageAttribute("Microsoft.Maintainability", "CA1502:AvoidExcessiveComplexity")]
public override string ConnectionString { get; set; }
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Maintainability", L"CA1502:AvoidExcessiveComplexity")
]
public:
virtual property String^ ConnectionString {
String^ get () override;
void set (String^ value) override;
}
```

J#

```
/** @property */
public String get_ConnectionString ()

/** @property */
public void set_ConnectionString (String value)
```

JScript

```
public override function get ConnectionString () : String
public override function set ConnectionString (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.Database Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Database As String
```

C#

```
public override string Database { get; }
```

C++

```
public:  
virtual property String^ Database {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_Database ()
```

JScript

```
public override function get Database () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.DataSource Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property DataSource As String
```

C#

```
public override string DataSource { get; }
```

C++

```
public:  
virtual property String^ DataSource {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_DataSource ()
```

JScript

```
public override function get DataSource () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.ServerVersion Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property ServerVersion As String
```

C#

```
public override string ServerVersion { get; }
```

C++

```
public:  
virtual property String^ ServerVersion {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_ServerVersion ()
```

JScript

```
public override function get ServerVersion () : String
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection.State Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides ReadOnly Property State As ConnectionState
```

C#

```
public override ConnectionState State { get; }
```

C++

```
public:  
virtual property ConnectionState State {  
    ConnectionState get () override;  
}
```

J#

```
/** @property */  
public ConnectionState get_State ()
```

JScript

```
public override function get State () : ConnectionState
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnection Class](#)

[SiebelConnection Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnection Events

Public Events

	Name	Description
	Disposed	(inherited from Component)
	StateChange	(inherited from DbConnection)

See Also

Reference

[SiebelConnection Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelConnectionException
    Inherits SiebelException
```

C#

```
[SerializableAttribute]
public class SiebelConnectionException : SiebelException
```

C++

```
[SerializableAttribute]
public ref class SiebelConnectionException : public SiebelException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelConnectionException extends SiebelException
```

JScript

```
SerializableAttribute
public class SiebelConnectionException extends SiebelException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

[Microsoft.Data.SiebelClient.SiebelException](#)

Microsoft.Data.SiebelClient.SiebelConnectionException

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Members

The following tables list the members exposed by the [SiebelConnectionException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelConnectionException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelConnectionException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HRESULT	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelConnectionException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Constructor

Overload List

Name	Description
SiebelConnectionException ()	
SiebelConnectionException (String)	
SiebelConnectionException (SerializationInfo, StreamingContext)	
SiebelConnectionException (String, Exception)	

See Also

Reference

[SiebelConnectionException Class](#)

[SiebelConnectionException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Constructor ()

Initializes a new instance of the [SiebelConnectionException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelConnectionException ()
```

C++

```
public:  
SiebelConnectionException ()
```

J#

```
public SiebelConnectionException ()
```

JScript

```
public function SiebelConnectionException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionException Class](#)

[SiebelConnectionException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelConnectionException (  
    string message  
)
```

C++

```
public:  
SiebelConnectionException (  
    String^ message  
)
```

J#

```
public SiebelConnectionException (  
    String message  
)
```

JScript

```
public function SiebelConnectionException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionException Class](#)

SiebelConnectionException Members
Microsoft.Data.SiebelClient Namespace

SiebelConnectionException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelConnectionException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelConnectionException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelConnectionException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelConnectionException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionException Class](#)

[SiebelConnectionException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public SiebelConnectionException (  
    string message,  
    Exception inner  
)
```

C++

```
public:  
SiebelConnectionException (  
    String^ message,  
    Exception^ inner  
)
```

J#

```
public SiebelConnectionException (  
    String message,  
    Exception inner  
)
```

JScript

```
public function SiebelConnectionException (  
    message : String,  
    inner : Exception  
)
```

Parameters

message

inner

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionException Class](#)

[SiebelConnectionException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelConnectionException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelConnectionException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SiebelConnectionStringBuilder  
    Inherits DbConnectionStringBuilder
```

C#

```
public sealed class SiebelConnectionStringBuilder : DbConnectionStringBuilder
```

C++

```
public ref class SiebelConnectionStringBuilder sealed : public DbConnectionStringBuilder
```

J#

```
public final class SiebelConnectionStringBuilder extends DbConnectionStringBuilder
```

JScript

```
public final class SiebelConnectionStringBuilder extends DbConnectionStringBuilder
```

Inheritance Hierarchy [System.Object](#)

[System.Data.Common.DbConnectionStringBuilder](#)

Microsoft.Data.SiebelClient.SiebelConnectionStringBuilder

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder Members

The following tables list the members exposed by the [SiebelConnectionStringBuilder](#) type.

Public Constructors

Name	Description
SiebelConnectionStringBuilder	

Top

Public Properties

Name	Description
 BrowsableConnectionString	(inherited from DbConnectionStringBuilder)
 Compression	
 ConnectionString	(inherited from DbConnectionStringBuilder)
 Count	(inherited from DbConnectionStringBuilder)
 Encryption	
 IsFixedSize	(inherited from DbConnectionStringBuilder)
 IsReadOnly	(inherited from DbConnectionStringBuilder)
 Item	(inherited from DbConnectionStringBuilder)
 Keys	Overridden.
 Language	
 Password	
 SiebelEnterpriseServer	
 SiebelGateway	
 SiebelObjectManager	
 SiebelRepository	
 SiebelServer	
 Transport	
 Username	
 Values	(inherited from DbConnectionStringBuilder)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 Add	(inherited from DbConnectionStringBuilder)
  AppendKeyValuePair	Overloaded. (inherited from DbConnectionStringBuilder)
 Clear	(inherited from DbConnectionStringBuilder)
 ContainsKey	(inherited from DbConnectionStringBuilder)
 Equals	Overloaded. (inherited from Object)
 EquivalentTo	(inherited from DbConnectionStringBuilder)
 GetHashCode	(inherited from Object)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 Remove	(inherited from DbConnectionStringBuilder)
 ShouldSerialize	(inherited from DbConnectionStringBuilder)
 ToString	(inherited from DbConnectionStringBuilder)
 TryGetValue	(inherited from DbConnectionStringBuilder)

Top

Protected Methods

Name	Description
 ClearPropertyDescriptors	(inherited from DbConnectionStringBuilder)
 Finalize	(inherited from Object)
 GetProperties	(inherited from DbConnectionStringBuilder)
 MemberwiseClone	(inherited from Object)

Top

Explicit Interface Implementations

Name	Description
System.Collections.ICollection.CopyTo	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Add	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Contains	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.GetEnumerator	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Remove	(inherited from DbConnectionStringBuilder)
System.Collections.IEnumerable.GetEnumerator	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetAttributes	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetClassName	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetComponentName	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetConverter	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetDefaultEvent	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetDefaultProperty	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetEditor	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetEvents	Overloaded. (inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetProperties	Overloaded. (inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetPropertyOwner	(inherited from DbConnectionStringBuilder)
System.Collections.ICollection.IsSynchronized	(inherited from DbConnectionStringBuilder)
System.Collections.ICollection.SyncRoot	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Item	(inherited from DbConnectionStringBuilder)

[Top](#)

[See Also](#)

Reference

[SiebelConnectionStringBuilder Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder Constructor

Initializes a new instance of the [SiebelConnectionStringBuilder](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelConnectionStringBuilder ()
```

C++

```
public:  
SiebelConnectionStringBuilder ()
```

J#

```
public SiebelConnectionStringBuilder ()
```

JScript

```
public function SiebelConnectionStringBuilder ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Add	(inherited from DbConnectionStringBuilder)
 	AppendKeyValuePair	Overloaded. (inherited from DbConnectionStringBuilder)
	Clear	(inherited from DbConnectionStringBuilder)
	ContainsKey	(inherited from DbConnectionStringBuilder)
	Equals	Overloaded. (inherited from Object)
	EquivalentTo	(inherited from DbConnectionStringBuilder)
	GetHashCode	(inherited from Object)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	Remove	(inherited from DbConnectionStringBuilder)
	ShouldSerialize	(inherited from DbConnectionStringBuilder)
	ToString	(inherited from DbConnectionStringBuilder)
	TryGetValue	(inherited from DbConnectionStringBuilder)

Protected Methods

	Name	Description
	ClearPropertyDescriptors	(inherited from DbConnectionStringBuilder)
	Finalize	(inherited from Object)
	GetProperties	(inherited from DbConnectionStringBuilder)
	MemberwiseClone	(inherited from Object)

Explicit Interface Implementations

Name	Description
System.Collections.ICollection.CopyTo	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Add	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Contains	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.GetEnumerator	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Remove	(inherited from DbConnectionStringBuilder)
System.Collections.IEnumerable.GetEnumerator	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetAttributes	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetClassName	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetComponentName	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetConverter	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetDefaultEvent	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetDefaultProperty	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetEditor	(inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetEvents	Overloaded. (inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetProperties	Overloaded. (inherited from DbConnectionStringBuilder)
System.ComponentModel.ICustomTypeDescriptor.GetPropertyOwner	(inherited from DbConnectionStringBuilder)

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder Properties

Public Properties

Name	Description
 BrowsableConnectionString	(inherited from DbConnectionStringBuilder)
 Compression	
 ConnectionString	(inherited from DbConnectionStringBuilder)
 Count	(inherited from DbConnectionStringBuilder)
 Encryption	
 IsFixedSize	(inherited from DbConnectionStringBuilder)
 IsReadOnly	(inherited from DbConnectionStringBuilder)
 Item	(inherited from DbConnectionStringBuilder)
 Keys	Overridden.
 Language	
 Password	
 SiebelEnterpriseServer	
 SiebelGateway	
 SiebelObjectManager	
 SiebelRepository	
 SiebelServer	
 Transport	
 Username	
 Values	(inherited from DbConnectionStringBuilder)

Explicit Interface Implementations

Name	Description
System.Collections.ICollection.IsSynchronized	(inherited from DbConnectionStringBuilder)
System.Collections.ICollection.SyncRoot	(inherited from DbConnectionStringBuilder)
System.Collections.IDictionary.Item	(inherited from DbConnectionStringBuilder)

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Compression Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Server")> _
<DescriptionAttribute("Type of compression to use for communication with server(optional)")
> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
Public Property Compression As String
```

C#

```
[CategoryAttribute("Server")]
[DescriptionAttribute("Type of compression to use for communication with server(optional)")
]
[RefreshPropertiesAttribute(RefreshProperties.All)]
public string Compression { get; set; }
```

C++

```
[CategoryAttribute(L"Server")]
[DescriptionAttribute(L"Type of compression to use for communication with server(optional)"
)]
[RefreshPropertiesAttribute(RefreshProperties::All)]
public:
property String^ Compression {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Compression ()

/** @property */
public void set_Compression (String value)
```

JScript

```
public function get Compression () : String
public function set Compression (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Encryption Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Type of encryption to use for communication with server(optional)")>
_
<CategoryAttribute("Server")> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
Public Property Encryption As String
```

C#

```
[DescriptionAttribute("Type of encryption to use for communication with server(optional)")]
[CategoryAttribute("Server")]
[RefreshPropertiesAttribute(RefreshProperties.All)]
public string Encryption { get; set; }
```

C++

```
[DescriptionAttribute(L"Type of encryption to use for communication with server(optional)")]
]
[CategoryAttribute(L"Server")]
[RefreshPropertiesAttribute(RefreshProperties::All)]
public:
property String^ Encryption {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Encryption ()

/** @property */
public void set_Encryption (String value)
```

JScript

```
public function get Encryption () : String
public function set Encryption (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Keys Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<BrowsableAttribute(False)> _
Public Overrides ReadOnly Property Keys As ICollection
```

C#

```
[BrowsableAttribute(false)]
public override ICollection Keys { get; }
```

C++

```
[BrowsableAttribute(false)]
public:
virtual property ICollection^ Keys {
    ICollection^ get () override;
}
```

J#

```
/** @property */
public ICollection get_Keys ()
```

JScript

```
public override function get Keys () : ICollection
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Language Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Language")> _
<CategoryAttribute("Server")> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
Public Property Language As String
```

C#

```
[DescriptionAttribute("Language")]
[CategoryAttribute("Server")]
[RefreshPropertiesAttribute(RefreshProperties.All)]
public string Language { get; set; }
```

C++

```
[DescriptionAttribute(L"Language")]
[CategoryAttribute(L"Server")]
[RefreshPropertiesAttribute(RefreshProperties::All)]
public:
property String^ Language {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Language ()

/** @property */
public void set_Language (String value)
```

JScript

```
public function get Language () : String

public function set Language (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Password Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Authetication")> _
<PasswordPropertyTextAttribute(True)> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
<DescriptionAttribute("Siebel User Password")> _
Public Property Password As String
```

C#

```
[CategoryAttribute("Authetication")]
[PasswordPropertyTextAttribute(true)]
[RefreshPropertiesAttribute(RefreshProperties.All)]
[DescriptionAttribute("Siebel User Password")]
public string Password { get; set; }
```

C++

```
[CategoryAttribute(L"Authetication")]
[PasswordPropertyTextAttribute(true)]
[RefreshPropertiesAttribute(RefreshProperties::All)]
[DescriptionAttribute(L"Siebel User Password")]
public:
property String^ Password {
    String^ get ();
    void set (String^ value);
}
```

J#

```
/** @property */
public String get_Password ()

/** @property */
public void set_Password (String value)
```

JScript

```
public function get Password () : String

public function set Password (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.SiebelEnterpriseServer Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<RefreshPropertiesAttribute(RefreshProperties.All)> _
<CategoryAttribute("Server")> _
<DescriptionAttribute("Siebel Enterprise Servername")> _
Public Property SiebelEnterpriseServer As String
```

C#

```
[RefreshPropertiesAttribute(RefreshProperties.All)]
[CategoryAttribute("Server")]
[DescriptionAttribute("Siebel Enterprise Servername")]
public string SiebelEnterpriseServer { get; set; }
```

C++

```
[RefreshPropertiesAttribute(RefreshProperties::All)]
[CategoryAttribute(L"Server")]
[DescriptionAttribute(L"Siebel Enterprise Servername")]
public:
property String^ SiebelEnterpriseServer {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelEnterpriseServer ()

/** @property */
public void set_SiebelEnterpriseServer (String value)
```

JScript

```
public function get SiebelEnterpriseServer () : String

public function set SiebelEnterpriseServer (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.SiebelGateway Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<RefreshPropertiesAttribute(RefreshProperties.All)> _
<CategoryAttribute("Server")> _
<DescriptionAttribute("Siebel server IP:Port")> _
Public Property SiebelGateway As String
```

C#

```
[RefreshPropertiesAttribute(RefreshProperties.All)]
[CategoryAttribute("Server")]
[DescriptionAttribute("Siebel server IP:Port")]
public string SiebelGateway { get; set; }
```

C++

```
[RefreshPropertiesAttribute(RefreshProperties::All)]
[CategoryAttribute(L"Server")]
[DescriptionAttribute(L"Siebel server IP:Port")]
public:
property String^ SiebelGateway {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelGateway ()

/** @property */
public void set_SiebelGateway (String value)
```

JScript

```
public function get SiebelGateway () : String

public function set SiebelGateway (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.SiebelObjectManager Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Server")> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
<DescriptionAttribute("Siebel Object Manager")> _
Public Property SiebelObjectManager As String
```

C#

```
[CategoryAttribute("Server")]
[RefreshPropertiesAttribute(RefreshProperties.All)]
[DescriptionAttribute("Siebel Object Manager")]
public string SiebelObjectManager { get; set; }
```

C++

```
[CategoryAttribute(L"Server")]
[RefreshPropertiesAttribute(RefreshProperties::All)]
[DescriptionAttribute(L"Siebel Object Manager")]
public:
property String^ SiebelObjectManager {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelObjectManager ()

/** @property */
public void set_SiebelObjectManager (String value)
```

JScript

```
public function get SiebelObjectManager () : String

public function set SiebelObjectManager (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.SiebelRepository Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Server")> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
<DescriptionAttribute("Siebel Repository(Optional)")> _
Public Property SiebelRepository As String
```

C#

```
[CategoryAttribute("Server")]
[RefreshPropertiesAttribute(RefreshProperties.All)]
[DescriptionAttribute("Siebel Repository(Optional)")]
public string SiebelRepository { get; set; }
```

C++

```
[CategoryAttribute(L"Server")]
[RefreshPropertiesAttribute(RefreshProperties::All)]
[DescriptionAttribute(L"Siebel Repository(Optional)")]
public:
property String^ SiebelRepository {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelRepository ()

/** @property */
public void set_SiebelRepository (String value)
```

JScript

```
public function get SiebelRepository () : String

public function set SiebelRepository (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.SiebelServer Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<RefreshPropertiesAttribute(RefreshProperties.All)> _
<CategoryAttribute("Server")> _
<DescriptionAttribute("Siebel Server(Optional needed for 7.5.x)")> _
Public Property SiebelServer As String
```

C#

```
[RefreshPropertiesAttribute(RefreshProperties.All)]
[CategoryAttribute("Server")]
[DescriptionAttribute("Siebel Server(Optional needed for 7.5.x)")]
public string SiebelServer { get; set; }
```

C++

```
[RefreshPropertiesAttribute(RefreshProperties::All)]
[CategoryAttribute(L"Server")]
[DescriptionAttribute(L"Siebel Server(Optional needed for 7.5.x)")]
public:
property String^ SiebelServer {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_SiebelServer ()

/** @property */
public void set_SiebelServer (String value)
```

JScript

```
public function get SiebelServer () : String

public function set SiebelServer (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Transport Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<DescriptionAttribute("Transport to be used by Siebel COM data control(optional)")> _
<CategoryAttribute("Server")> _
<RefreshPropertiesAttribute(RefreshProperties.All)> _
Public Property Transport As String
```

C#

```
[DescriptionAttribute("Transport to be used by Siebel COM data control(optional)")]
[CategoryAttribute("Server")]
[RefreshPropertiesAttribute(RefreshProperties.All)]
public string Transport { get; set; }
```

C++

```
[DescriptionAttribute(L"Transport to be used by Siebel COM data control(optional)")]
[CategoryAttribute(L"Server")]
[RefreshPropertiesAttribute(RefreshProperties::All)]
public:
property String^ Transport {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Transport ()

/** @property */
public void set_Transport (String value)
```

JScript

```
public function get Transport () : String

public function set Transport (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelConnectionStringBuilder.Username Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<RefreshPropertiesAttribute(RefreshProperties.All)> _
<CategoryAttribute("Authetication")> _
<DescriptionAttribute("Siebel Username")> _
Public Property Username As String
```

C#

```
[RefreshPropertiesAttribute(RefreshProperties.All)]
[CategoryAttribute("Authetication")]
[DescriptionAttribute("Siebel Username")]
public string Username { get; set; }
```

C++

```
[RefreshPropertiesAttribute(RefreshProperties::All)]
[CategoryAttribute(L"Authetication")]
[DescriptionAttribute(L"Siebel Username")]
public:
property String^ Username {
String^ get ();
void set (String^ value);
}
```

J#

```
/** @property */
public String get_Username ()

/** @property */
public void set_Username (String value)
```

JScript

```
public function get Username () : String

public function set Username (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with

SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelConnectionStringBuilder Class](#)

[SiebelConnectionStringBuilder Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelDataReaderException
    Inherits SiebelException
```

C#

```
[SerializableAttribute]
public class SiebelDataReaderException : SiebelException
```

C++

```
[SerializableAttribute]
public ref class SiebelDataReaderException : public SiebelException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelDataReaderException extends SiebelException
```

JScript

```
SerializableAttribute
public class SiebelDataReaderException extends SiebelException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

[Microsoft.Data.SiebelClient.SiebelException](#)

Microsoft.Data.SiebelClient.SiebelDataReaderException

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelDataReaderException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Members

The following tables list the members exposed by the [SiebelDataReaderException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelDataReaderException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelDataReaderException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HResult	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelDataReaderException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Constructor

Overload List

Name	Description
SiebelDataReaderException ()	
SiebelDataReaderException (String)	
SiebelDataReaderException (SerializationInfo, StreamingContext)	
SiebelDataReaderException (String, Exception)	

See Also

Reference

[SiebelDataReaderException Class](#)

[SiebelDataReaderException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Constructor ()

Initializes a new instance of the [SiebelDataReaderException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelDataReaderException ()
```

C++

```
public:  
SiebelDataReaderException ()
```

J#

```
public SiebelDataReaderException ()
```

JScript

```
public function SiebelDataReaderException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelDataReaderException Class](#)

[SiebelDataReaderException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelDataReaderException (  
    string message  
)
```

C++

```
public:  
SiebelDataReaderException (  
    String^ message  
)
```

J#

```
public SiebelDataReaderException (  
    String message  
)
```

JScript

```
public function SiebelDataReaderException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelDataReaderException Class](#)

SiebelDataReaderException Members
Microsoft.Data.SiebelClient Namespace

SiebelDataReaderException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelDataReaderException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelDataReaderException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelDataReaderException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelDataReaderException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelDataReaderException Class](#)

[SiebelDataReaderException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public SiebelDataReaderException (  
    string message,  
    Exception inner  
)
```

C++

```
public:  
SiebelDataReaderException (  
    String^ message,  
    Exception^ inner  
)
```

J#

```
public SiebelDataReaderException (  
    String message,  
    Exception inner  
)
```

JScript

```
public function SiebelDataReaderException (  
    message : String,  
    inner : Exception  
)
```

Parameters

message

inner

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelDataReaderException Class](#)

[SiebelDataReaderException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelDataReaderException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelDataReaderException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelDataReaderException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelException
    Inherits DbException
```

C#

```
[SerializableAttribute]
public class SiebelException : DbException
```

C++

```
[SerializableAttribute]
public ref class SiebelException : public DbException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelException extends DbException
```

JScript

```
SerializableAttribute
public class SiebelException extends DbException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

Microsoft.Data.SiebelClient.SiebelException

[Microsoft.Data.SiebelClient.SiebelCommandException](#)

[Microsoft.Data.SiebelClient.SiebelConnectionException](#)

[Microsoft.Data.SiebelClient.SiebelDataReaderException](#)

[Microsoft.Data.SiebelClient.SiebelParameterException](#)

[Microsoft.Data.SiebelClient.SiebelParserException](#)

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Members

The following tables list the members exposed by the [SiebelException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HResult	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Constructor

Overload List

Name	Description
SiebelException ()	
SiebelException (String)	
SiebelException (SerializationInfo, StreamingContext)	
SiebelException (String, Exception)	

See Also

Reference

[SiebelException Class](#)

[SiebelException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Constructor ()

Initializes a new instance of the [SiebelException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelException ()
```

C++

```
public:  
SiebelException ()
```

J#

```
public SiebelException ()
```

JScript

```
public function SiebelException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelException Class](#)

[SiebelException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelException (  
    string message  
)
```

C++

```
public:  
SiebelException (  
    String^ message  
)
```

J#

```
public SiebelException (  
    String message  
)
```

JScript

```
public function SiebelException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelException Class](#)

SiebelException Members
Microsoft.Data.SiebelClient Namespace

SiebelException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelException Class](#)

[SiebelException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    innerException As Exception _  
)
```

C#

```
public SiebelException (  
    string message,  
    Exception innerException  
)
```

C++

```
public:  
SiebelException (  
    String^ message,  
    Exception^ innerException  
)
```

J#

```
public SiebelException (  
    String message,  
    Exception innerException  
)
```

JScript

```
public function SiebelException (  
    message : String,  
    innerException : Exception  
)
```

Parameters

message

innerException

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelException Class](#)

[SiebelException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Class SiebelParameter  
    Inherits DbParameter
```

C#

```
public class SiebelParameter : DbParameter
```

C++

```
public ref class SiebelParameter : public DbParameter
```

J#

```
public class SiebelParameter extends DbParameter
```

JScript

```
public class SiebelParameter extends DbParameter
```

Inheritance Hierarchy [System.Object](#)

[System.MarshalByRefObject](#)

[System.Data.Common.DbParameter](#)

Microsoft.Data.SiebelClient.SiebelParameter

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Members

The following tables list the members exposed by the [SiebelParameter](#) type.

Public Constructors

Name	Description
SiebelParameter	Overloaded.

Top

Public Properties

Name	Description
 DbType	Overridden.
 Direction	Overridden.
 IsNullable	Overridden.
 ParameterName	Overridden.
 Size	Overridden.
 SourceColumn	Overridden.
 SourceColumnNullMapping	Overridden.
 SourceVersion	Overridden.
 Value	Overridden.

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 CreateObjRef	(inherited from MarshalByRefObject)
 Equals	Overloaded. (inherited from Object)
 GetHashCode	(inherited from Object)
 GetLifetimeService	(inherited from MarshalByRefObject)
 GetType	(inherited from Object)
 InitializeLifetimeService	(inherited from MarshalByRefObject)
  ReferenceEquals	(inherited from Object)
 ResetDbType	Overridden.
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)

Top

Explicit Interface Implementations

Name	Description
System.Data.DbDataParameter.Precision	(inherited from DbParameter)
System.Data.DbDataParameter.Scale	(inherited from DbParameter)

Top

See Also

Reference

[SiebelParameter Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Constructor

Overload List

Name	Description
SiebelParameter ()	
SiebelParameter (String)	
SiebelParameter (String, Object)	

See Also

Reference

[SiebelParameter Class](#)

[SiebelParameter Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Constructor ()

Initializes a new instance of the [SiebelParameter](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelParameter ()
```

C++

```
public:  
SiebelParameter ()
```

J#

```
public SiebelParameter ()
```

JScript

```
public function SiebelParameter ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

[SiebelParameter Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    parameter As String _  
)
```

C#

```
public SiebelParameter (  
    string parameter  
)
```

C++

```
public:  
SiebelParameter (  
    String^ parameter  
)
```

J#

```
public SiebelParameter (  
    String parameter  
)
```

JScript

```
public function SiebelParameter (  
    parameter : String  
)
```

Parameters

parameter

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter Constructor (String, Object)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    parameter As String, _  
    value As Object _  
)
```

C#

```
public SiebelParameter (  
    string parameter,  
    Object value  
)
```

C++

```
public:  
SiebelParameter (  
    String^ parameter,  
    Object^ value  
)
```

J#

```
public SiebelParameter (  
    String parameter,  
    Object value  
)
```

JScript

```
public function SiebelParameter (  
    parameter : String,  
    value : Object  
)
```

Parameters

parameter

value

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

[SiebelParameter Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	CreateObjRef	(inherited from MarshalByRefObject)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetLifetimeService	(inherited from MarshalByRefObject)
	GetType	(inherited from Object)
	InitializeLifetimeService	(inherited from MarshalByRefObject)
 	ReferenceEquals	(inherited from Object)
	ResetDbType	Overridden.
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	Overloaded. (inherited from MarshalByRefObject)

See Also

Reference

[SiebelParameter Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter.ResetDbType Method

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Overrides Sub ResetDbType
```

C#

```
public override void ResetDbType ()
```

C++

```
public:  
virtual void ResetDbType () override
```

J#

```
public void ResetDbType ()
```

JScript

```
public override function ResetDbType ()
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

[SiebelParameter Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter Properties

Public Properties

	Name	Description
	DbType	Overridden.
	Direction	Overridden.
	IsNullable	Overridden.
	ParameterName	Overridden.
	Size	Overridden.
	SourceColumn	Overridden.
	SourceColumnNullMapping	Overridden.
	SourceVersion	Overridden.
	Value	Overridden.

Explicit Interface Implementations

Name	Description
System.Data.IDbDataParameter.Precision	(inherited from DbParameter)
System.Data.IDbDataParameter.Scale	(inherited from DbParameter)

See Also

Reference

[SiebelParameter Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameter.DbType Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property DbType As DbType
```

C#

```
public override DbType DbType { get; set; }
```

C++

```
public:
virtual property DbType DbType {
    DbType get () override;
    void set (DbType value) override;
}
```

J#

```
/** @property */
public DbType get_DbType ()

/** @property */
public void set_DbType (DbType value)
```

JScript

```
public override function get DbType () : DbType

public override function set DbType (value : DbType)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.Direction Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property Direction As ParameterDirection
```

C#

```
public override ParameterDirection Direction { get; set; }
```

C++

```
public:
virtual property ParameterDirection Direction {
    ParameterDirection get () override;
    void set (ParameterDirection value) override;
}
```

J#

```
/** @property */
public ParameterDirection get_Direction ()

/** @property */
public void set_Direction (ParameterDirection value)
```

JScript

```
public override function get Direction () : ParameterDirection

public override function set Direction (value : ParameterDirection)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.IsNullable Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property IsNullable As Boolean
```

C#

```
public override bool IsNullable { get; set; }
```

C++

```
public:  
virtual property bool IsNullable {  
    bool get () override;  
    void set (bool value) override;  
}
```

J#

```
/** @property */  
public boolean get_IsNullable ()  
  
/** @property */  
public void set_IsNullable (boolean value)
```

JScript

```
public override function get IsNullable () : boolean  
  
public override function set IsNullable (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.ParameterName Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration
Public Overrides Property ParameterName As String
```

C#

```
public override string ParameterName { get; set; }
```

C++

```
public:
virtual property String^ ParameterName {
String^ get () override;
void set (String^ value) override;
}
```

J#

```
/** @property */
public String get_ParameterName ()

/** @property */
public void set_ParameterName (String value)
```

JScript

```
public override function get ParameterName () : String
public override function set ParameterName (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.Size Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property Size As Integer
```

C#

```
public override int Size { get; set; }
```

C++

```
public:  
virtual property int Size {  
    int get () override;  
    void set (int value) override;  
}
```

J#

```
/** @property */  
public int get_Size ()  
  
/** @property */  
public void set_Size (int value)
```

JScript

```
public override function get Size () : int  
  
public override function set Size (value : int)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.SourceColumn Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property SourceColumn As String
```

C#

```
public override string SourceColumn { get; set; }
```

C++

```
public:
virtual property String^ SourceColumn {
String^ get () override;
void set (String^ value) override;
}
```

J#

```
/** @property */
public String get_SourceColumn ()

/** @property */
public void set_SourceColumn (String value)
```

JScript

```
public override function get SourceColumn () : String

public override function set SourceColumn (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.SourceColumnNullMapping Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Property SourceColumnNullMapping As Boolean
```

C#

```
public override bool SourceColumnNullMapping { get; set; }
```

C++

```
public:  
virtual property bool SourceColumnNullMapping {  
    bool get () override;  
    void set (bool value) override;  
}
```

J#

```
/** @property */  
public boolean get_SourceColumnNullMapping ()  
  
/** @property */  
public void set_SourceColumnNullMapping (boolean value)
```

JScript

```
public override function get SourceColumnNullMapping () : boolean  
  
public override function set SourceColumnNullMapping (value : boolean)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.SourceVersion Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property SourceVersion As DataRowVersion
```

C#

```
public override DataRowVersion SourceVersion { get; set; }
```

C++

```
public:
virtual property DataRowVersion SourceVersion {
    DataRowVersion get () override;
    void set (DataRowVersion value) override;
}
```

J#

```
/** @property */
public DataRowVersion get_SourceVersion ()

/** @property */
public void set_SourceVersion (DataRowVersion value)
```

JScript

```
public override function get SourceVersion () : DataRowVersion

public override function set SourceVersion (value : DataRowVersion)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameter.Value Property

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

Public Overrides Property Value As Object
```

C#

```
public override Object Value { get; set; }
```

C++

```
public:
virtual property Object^ Value {
    Object^ get () override;
    void set (Object^ value) override;
}
```

J#

```
/** @property */
public Object get_Value ()

/** @property */
public void set_Value (Object value)
```

JScript

```
public override function get Value () : Object

public override function set Value (value : Object)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameter Class](#)

SiebelParameter Members
Microsoft.Data.SiebelClient Namespace

SiebelParameterException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelParameterException
    Inherits SiebelException
```

C#

```
[SerializableAttribute]
public class SiebelParameterException : SiebelException
```

C++

```
[SerializableAttribute]
public ref class SiebelParameterException : public SiebelException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelParameterException extends SiebelException
```

JScript

```
SerializableAttribute
public class SiebelParameterException extends SiebelException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

[Microsoft.Data.SiebelClient.SiebelException](#)

Microsoft.Data.SiebelClient.SiebelParameterException

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameterException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Members

The following tables list the members exposed by the [SiebelParameterException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelParameterException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelParameterException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HResult	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelParameterException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Constructor

Overload List

Name	Description
SiebelParameterException ()	
SiebelParameterException (String)	
SiebelParameterException (SerializationInfo, StreamingContext)	
SiebelParameterException (String, Exception)	

See Also

Reference

[SiebelParameterException Class](#)

[SiebelParameterException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Constructor ()

Initializes a new instance of the [SiebelParameterException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelParameterException ()
```

C++

```
public:  
SiebelParameterException ()
```

J#

```
public SiebelParameterException ()
```

JScript

```
public function SiebelParameterException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameterException Class](#)

[SiebelParameterException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelParameterException (  
    string message  
)
```

C++

```
public:  
SiebelParameterException (  
    String^ message  
)
```

J#

```
public SiebelParameterException (  
    String message  
)
```

JScript

```
public function SiebelParameterException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameterException Class](#)

SiebelParameterException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelParameterException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelParameterException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelParameterException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelParameterException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameterException Class](#)

[SiebelParameterException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public SiebelParameterException (  
    string message,  
    Exception inner  
)
```

C++

```
public:  
SiebelParameterException (  
    String^ message,  
    Exception^ inner  
)
```

J#

```
public SiebelParameterException (  
    String message,  
    Exception inner  
)
```

JScript

```
public function SiebelParameterException (  
    message : String,  
    inner : Exception  
)
```

Parameters

message

inner

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParameterException Class](#)

[SiebelParameterException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelParameterException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParameterException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelParameterException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Class

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration

<SerializableAttribute> _
Public Class SiebelParserException
    Inherits SiebelException
```

C#

```
[SerializableAttribute]
public class SiebelParserException : SiebelException
```

C++

```
[SerializableAttribute]
public ref class SiebelParserException : public SiebelException
```

J#

```
/** @attribute SerializableAttribute() */
public class SiebelParserException extends SiebelException
```

JScript

```
SerializableAttribute
public class SiebelParserException extends SiebelException
```

Inheritance Hierarchy [System.Object](#)

[System.Exception](#)

[System.SystemException](#)

[System.Runtime.InteropServices.ExternalException](#)

[System.Data.Common.DbException](#)

[Microsoft.Data.SiebelClient.SiebelException](#)

Microsoft.Data.SiebelClient.SiebelParserException

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParserException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Members

The following tables list the members exposed by the [SiebelParserException](#) type.

Public Constructors (see also [Protected Constructors](#))

Name	Description
SiebelParserException	Overloaded.

[Top](#)

Protected Constructors

Name	Description
SiebelParserException	Overloaded.

[Top](#)

Public Properties (see also [Protected Properties](#))

Name	Description
 Data	(inherited from Exception)
 ErrorCode	(inherited from ExternalException)
 HelpLink	(inherited from Exception)
 InnerException	(inherited from Exception)
 Message	(inherited from Exception)
 Source	(inherited from Exception)
 StackTrace	(inherited from Exception)
 TargetSite	(inherited from Exception)

[Top](#)

Protected Properties

Name	Description
 HResult	(inherited from Exception)

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetBaseException	(inherited from Exception)
 GetHashCode	(inherited from Object)
 GetObjectData	(inherited from Exception)
 GetType	(inherited from Exception)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from Exception)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SiebelParserException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Constructor

Overload List

Name	Description
SiebelParserException ()	
SiebelParserException (String)	
SiebelParserException (SerializationInfo, StreamingContext)	
SiebelParserException (String, Exception)	

See Also

Reference

[SiebelParserException Class](#)

[SiebelParserException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Constructor ()

Initializes a new instance of the [SiebelParserException](#) Class.

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SiebelParserException ()
```

C++

```
public:  
SiebelParserException ()
```

J#

```
public SiebelParserException ()
```

JScript

```
public function SiebelParserException ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParserException Class](#)

[SiebelParserException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Constructor (String)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String _  
)
```

C#

```
public SiebelParserException (  
    string message  
)
```

C++

```
public:  
SiebelParserException (  
    String^ message  
)
```

J#

```
public SiebelParserException (  
    String message  
)
```

JScript

```
public function SiebelParserException (  
    message : String  
)
```

Parameters

message

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParserException Class](#)

SiebelParserException Members
Microsoft.Data.SiebelClient Namespace

SiebelParserException Constructor (SerializationInfo, StreamingContext)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Protected Sub New ( _  
    info As SerializationInfo, _  
    context As StreamingContext _  
)
```

C#

```
protected SiebelParserException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

C++

```
protected:  
SiebelParserException (  
    SerializationInfo^ info,  
    StreamingContext context  
)
```

J#

```
protected SiebelParserException (  
    SerializationInfo info,  
    StreamingContext context  
)
```

JScript

```
protected function SiebelParserException (  
    info : SerializationInfo,  
    context : StreamingContext  
)
```

Parameters

info

context

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParserException Class](#)

[SiebelParserException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Constructor (String, Exception)

Namespace: Microsoft.Data.SiebelClient

Assembly: Microsoft.Data.SiebelClient (in microsoft.data.siebelclient.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    message As String, _  
    inner As Exception _  
)
```

C#

```
public SiebelParserException (  
    string message,  
    Exception inner  
)
```

C++

```
public:  
SiebelParserException (  
    String^ message,  
    Exception^ inner  
)
```

J#

```
public SiebelParserException (  
    String message,  
    Exception inner  
)
```

JScript

```
public function SiebelParserException (  
    message : String,  
    inner : Exception  
)
```

Parameters

message

inner

Platforms

Development Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

Target Platforms

The 32-bit edition of Windows Server 2003 R2 (except the Web Editions); The 32-bit edition of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit edition of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2

See Also

Reference

[SiebelParserException Class](#)

[SiebelParserException Members](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetBaseException	(inherited from Exception)
	GetHashCode	(inherited from Object)
	GetObjectData	(inherited from Exception)
	GetType	(inherited from Exception)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Exception)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SiebelParserException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

SiebelParserException Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	Data	(inherited from Exception)
	ErrorCode	(inherited from ExternalException)
	HelpLink	(inherited from Exception)
	InnerException	(inherited from Exception)
	Message	(inherited from Exception)
	Source	(inherited from Exception)
	StackTrace	(inherited from Exception)
	TargetSite	(inherited from Exception)

Protected Properties

	Name	Description
	HResult	(inherited from Exception)

See Also

Reference

[SiebelParserException Class](#)

[Microsoft.Data.SiebelClient Namespace](#)

Microsoft BizTalk Adapter for SQL Server Help

Welcome to Microsoft BizTalk Adapter for SQL Server Help, which includes a variety of resources that can help you learn to develop, deploy, administer, and use the SQL adapter.

To download a copy of this Help, go to <http://go.microsoft.com/fwlink/?LinkId=188969>.

	<h2>Getting Started</h2> <p>Information about new features, limitations, prerequisites, and topics for users who are new to the adapter.</p>
	<h2>SQL Adapter Tutorials</h2> <p>Step-by-step instructions for learning how to use the adapter for specific scenarios.</p>
	<h2>Planning and Architecture</h2> <p>Architectural overview of the adapter.</p>
	<h2>Development</h2> <p>Instructions on how to use the adapter with BizTalk Server, the Windows Communication Foundation (WCF) service model, and the WCF channel model.</p>
	<h2>Deployment</h2> <p>Information about installing the adapter, migrating from previous versions of the adapter, and troubleshooting the installation.</p>
	<h2>Troubleshooting</h2> <p>Information about enabling tracing and about troubleshooting issues with the BizTalk Adapter Pack and with the adapter, in particular.</p>
	<h2>Samples</h2> <p>Samples demonstrating how to use the adapter.</p>
	<h2>Community Resources</h2> <p>Information about how to get assistance with the BizTalk Adapter Pack.</p>

Getting Started

Important

If this document is installed as part of the SQL adapter installation (by running the SQLAdapterSetup.msi), you must ignore all references to Microsoft BizTalk Adapter Pack. Also, all references to the BizTalk Adapter Pack Setup should be interpreted as the SQL adapter Setup.

This section provides an overview of the adapter, prerequisites, and topics for users who are new to Microsoft BizTalk Adapter Pack. It discusses the features of Microsoft BizTalk Adapter for SQL Server and the different operations that can be performed on the SQL Server database by using the adapter.

What is an adapter? An adapter is a software component that enables you to send and receive messages to and from a line-of-business (LOB) system. The primary design goal of an adapter is to facilitate the exchange of business documents between trading partners. Because each business system may adhere to specific document formats and protocols, the adapter must use a delivery mechanism that conforms to commonly recognized standards and protocols to provide a uniform interface to the users.

The adapters in the BizTalk Adapter Pack can be divided into two broad categories:

- **LOB adapters.** Such adapters provide service-oriented programming model to access LOB systems—for example, adapters for SAP or Siebel.
- **Data adapters.** Such adapters provide service-oriented programming model to access databases—for example, adapters for the Oracle database or SQL Server.

There are five adapters in the BizTalk Adapter Pack:

- Microsoft BizTalk Adapter for SQL Server (SQL adapter).

Note

The SQL adapter is also available outside the BizTalk Adapter Pack as a separate adapter.

- Microsoft BizTalk Adapter for Oracle E-Business Suite (Oracle E-Business adapter).
- Microsoft BizTalk Adapter for Oracle Database (Oracle Database adapter).
- Microsoft BizTalk Adapter for mySAP Business Suite (SAP adapter).
- Microsoft BizTalk Adapter for Siebel eBusiness Applications (Siebel adapter).

Note

The Siebel adapter is not available for 64-bit platforms.

If you do not already know how you want to use the SQL adapter at your company, it is recommended that you start by exploring the features and functionality of the adapter described in [Understanding BizTalk Adapter for SQL Server](#).

In This Section

- [Prerequisite Skills and Knowledge](#)
- [Common Administrative Tasks](#)
- [Common Developer Tasks](#)

- [Understanding BizTalk Adapter for SQL Server](#)
- [SQL Adapter Tutorials](#)
- [Community Resources](#)
- [Frequently Asked Questions](#)

Prerequisite Skills and Knowledge

The potential users for the BizTalk Adapter Pack are:

- Developers who directly program to the adapters.
- IT professionals who consume the adapters using other integration platforms like Microsoft BizTalk Server and other tools.
- Independent software vendors (ISVs) who build solutions on top of the adapters.

The prerequisite skills and knowledge required for each role are detailed below.

Prerequisites for Developers

A developer using Microsoft BizTalk Adapter for SQL Server must be at least moderately experienced with the use of:

- Microsoft Visual Studio 2010 and the development of .NET solutions
- Programming with the .NET Framework 3.5 SP1 and Microsoft .NET Framework 4

Note

.NET Framework 4 is essential to use BizTalk Adapter Pack with BizTalk Server 2010. However, you can use .NET Framework 3.5 SP1 and Microsoft .NET Framework 4 to write .NET applications (channel/proxy) for connecting using adapters.

- Programming with the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
- Extensible Markup Language (XML)
- XML Schema Definition (XSD) language
- Web Services Definition Language (WSDL)

Prerequisites for IT Professional

An IT professional using the Microsoft BizTalk Adapter for SQL Server must be at least moderately experienced with the use of:

- Microsoft BizTalk Server 2010
- Windows SharePoint Services

Prerequisites for ISVs

ISVs using the Microsoft BizTalk Adapter for SQL Server must be at least moderately experienced with the use of:

- The internal workings and concepts of SQL Server to be able to build applications on top of the adapters
- .NET Framework 3.5 SP1 and Microsoft .NET Framework 4
- The WCF LOB Adapter SDK

See Also

Other Resources

[Getting Started](#)

[Understanding BizTalk Adapter for SQL Server](#)

Common Administrative Tasks

The following table provides links to topics helpful for deploying, using, and maintaining Microsoft BizTalk Adapter for SQL Server.

Administrative Tasks <ul style="list-style-type: none">• SQL Adapter Tutorials• Deployment	Planning and Architecture <ul style="list-style-type: none">• Planning and Architecture• Understanding BizTalk Adapter for SQL Server• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Glossary
--	---

See Also

Concepts

[Common Developer Tasks](#)

Other Resources

[Getting Started](#)

Common Developer Tasks

The following table provides links to topics useful for developing applications on top of Microsoft BizTalk Adapter for SQL Server.

Developer Tasks <ul style="list-style-type: none">• Establishing a Connection to SQL Server• Developing BizTalk Applications• Developing Applications by Using the WCF Service Model• Developing Applications by Using the WCF Channel Model	Planning and Architecture <ul style="list-style-type: none">• Planning and Architecture• Understanding BizTalk Adapter for SQL Server• Security and Protection Other Resources <ul style="list-style-type: none">• Community Resources• Troubleshooting• Glossary
--	---

See Also

Concepts

[Common Administrative Tasks](#)

Other Resources

[Getting Started](#)

Understanding BizTalk Adapter for SQL Server

The BizTalk Adapter Pack enables service-oriented programmatic access making it possible to interact with an external system. The adapters provide the following advantages to clients:

- **Consistent design-time experience.** The adapters provide a common and user-friendly design-time experience for browsing, searching, and retrieving metadata of LOB artifacts.
- **Varied programming options.** The adapters provide a choice of programming model including the Windows Communication Foundation (WCF) channel model, WCF service model, ADO.NET, Web services, or BizTalk supported models.
- **Uniform experience across LOBs.** The adapters standardize on using WCF and the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, and hence provide a uniform experience of gaining access to any LOB system.

As mentioned, the adapters are built on top of the WCF LOB Adapter SDK. The WCF LOB Adapter SDK provides a common basis for building integration adapters that a variety of client applications such as BizTalk Server and Microsoft Office can consume. The WCF LOB Adapter SDK aligns the adapter strategy with the Microsoft Services strategy by exposing integration adapters as Windows Communication Foundation (WCF) channels. For more information about the WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation. The WCF LOB Adapter SDK documentation is installed along with the WCF LOB Adapter SDK, typically under <installation drive>\Program Files\WCF LOB Adapter SDK\Documents.

To perform operations on a SQL Server database, adapter clients must have access to relevant tables, procedures, views, scalar functions, and table valued functions. Database tables are the basic unit of storage in the SQL Server database. External applications can select, insert, delete, and update data from a table by using SQL statements. Applications can also access data in the tables by using procedures, views, scalar functions, and table valued functions. With Microsoft BizTalk Adapter for SQL Server, adapter clients can browse artifacts such as tables, procedures, views, and other such items in a SQL Server database. Adapter clients can select the artifacts they require for their solution, and retrieve metadata for those artifacts. This enables users to access and execute the operations on the artifacts in the SQL Server database.

This section lists the features of the SQL adapter.

In This Section

- [Overview of BizTalk Adapter for SQL Server](#)
- [New Features in BizTalk Adapter for SQL Server](#)
- [Limitations of BizTalk Adapter for SQL Server](#)

See Also

Other Resources

[Getting Started](#)

Overview of BizTalk Adapter for SQL Server

The Microsoft BizTalk Adapter for SQL Server exposes the SQL Server database as a WCF service. Adapter clients can perform operations on the SQL Server database by exchanging SOAP messages with the adapter. The adapter consumes the SOAP message and makes appropriate ADO.NET calls to perform the operation. The adapter returns the response from the SQL Server database back to the client in the form of SOAP messages.

The SQL adapter surfaces metadata of artifacts in the SQL Server database (such as tables, views, and procedures). This metadata describes the structure of a SOAP message in the form of Web Services Description Language (WSDL). The SQL adapter uses the Add Adapter Service Reference Visual Studio Plug-in, Add Adapter Metadata Wizard, and Consume Adapter Service BizTalk Project Add-in to enable adapter clients to retrieve metadata for operations. The adapter also generates programming artifacts that can be used in your programming solution. For more information about Add Adapter Service Reference Visual Studio Plug-in, Add Adapter Metadata Wizard, and Consume Adapter Service BizTalk Project Add-in, see [Connecting to SQL Server in Visual Studio](#).

The SQL adapter uses ADO.NET to communicate with the SQL Server database. You can use the SQL adapter to communicate with the SQL Server database in the following ways:

- By developing BizTalk applications. For more information, see [Developing BizTalk Applications](#).
- By using the Windows Communication Foundation (WCF) service model. For more information, see [Developing Applications by Using the WCF Service Model](#).
- By using the WCF channel model. For more information, see [Developing Applications by Using the WCF Channel Model](#).

In This Section

- [How Does the Adapter Connect to SQL Server?](#)
- [How Does the Adapter Surface SQL Server Metadata?](#)
- [What Operations Can Be Performed Using the Adapter?](#)
- [Other Features Supported by the Adapter](#)

See Also

Other Resources

[Understanding BizTalk Adapter for SQL Server](#)

How Does the Adapter Connect to SQL Server?

The Microsoft BizTalk Adapter for SQL Server requires adapter clients to provide a connection string, called the connection Uniform Resource Identifier (URI), to connect to the SQL Server database. Internally, the SQL adapter maps the URI to a database connection string to connect to the SQL Server database. With a connection URI, adapter clients can specify connection parameters to connect to an external system. For more information about the connection URI, see [SQL Server Connection URI](#).

In the connection URI, you can also specify the name of a failover SQL Server database on a standby computer to connect to if the primary SQL Server database is not available. The failover SQL Server database must be a mirror of the primary SQL Server database. The failover SQL Server database is specified using an optional parameter, FailoverPartner, in the connection URI. Providing a failover SQL Server database ensures high availability and data redundancy. For more information about high availability with respect to SQL Server, see "Using Database Mirroring" at <http://go.microsoft.com/fwlink/?LinkID=196852>.

Make sure you comply with the security guidelines when establishing a connection with the SQL Server database. For more information about security guidelines, see [Security and Protection](#).

See Also

Other Resources

[Overview of BizTalk Adapter for SQL Server](#)

[Establishing a Connection to SQL Server](#)

How Does the Adapter Surface SQL Server Metadata?

The metadata that Microsoft BizTalk Adapter for SQL Server surfaces from the SQL Server database describes the message structure for communicating with the SQL Server database using the adapter. The SQL adapter supports two interfaces for retrieving metadata.

- MetadataExchange provided by Windows Communication Foundation (WCF). WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from the SQL Server database.
- IMetadataRetrievalContract provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

The SQL adapter surfaces the SQL Server database artifacts and respective operations that the adapter clients can invoke. The adapter also surfaces operations (such as ExecuteNonQuery, ExecuteReader, and ExecuteScalar) that can be used to perform specific operations on the SQL Server database. These operations are discussed later in this topic.

Note

The SQL adapter surfaces artifacts in all the schemas in the SQL Server database that the currently connected user has access to. This implies that apart from the default schema (dbo), the adapter clients can also perform operations on artifacts in other schemas in the SQL Server database provided that the user credentials used to connect using the SQL adapter has access to those schemas in the SQL Server database. For information about a schema in SQL Server database, see <http://go.microsoft.com/fwlink/?LinkId=130148>.

You can use the adapter clients to browse, search, and retrieve metadata by:

- Creating a BizTalk project in Visual Studio
- Using the WCF service model
- Using the WCF channel model

When using a BizTalk project, you must use the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard to generate metadata for the operations that you want to perform on the SQL Server database. When using the WCF service model, you must use the Add Adapter Service Reference Visual Studio Plug-in to generate the proxy classes for performing operations on the SQL Server database. For more information about browsing, searching, and retrieving metadata using Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#).

Browsing Metadata

The SQL adapter enables adapter clients to browse database tables, views, stored procedures, and functions that are available in the SQL Server database. As part of the metadata browse operation, the adapter also surfaces the operations that can be performed on the SQL Server database, including some custom operations supported by the adapters. These operations are available from Consume Adapter Service Add-in, Add Adapter Metadata Wizard or Add Adapter Service Reference Plug-in. The SQL adapter surfaces the following operations:

- The operations on tables, views, procedures, scalar functions, and table valued functions. For example, the SQL adapter may surface Insert, Update, Select, and Delete operations for the EMPLOYEE table.
- The Set<column name> operation for tables and views that enables adapter clients to write large data values in a streaming fashion. The Set operation is only returned for those tables and views that contain columns with any of the following data types: Varchar(Max), Nvarchar(Max) or Varbinary(Max). For more information, see [Operations on Tables and Views That Contain Large Data Types](#).
- The ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations that enable adapter clients to execute arbitrary SQL statements in SQL Server. For more information about these operations, see [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

- The Polling and Notification operations to receive inbound messages from SQL Server. For information about the Polling operation, see [Support for Polling](#); for information about the Notification operation, see [Receiving Query Notifications](#).

For more information about how the metadata is categorized, see [Metadata Node IDs](#).

Searching Metadata

With the SQL adapter, it is possible to perform a search query on the SQL Server database by using the SQL Server search expressions that are compatible with the LIKE operator. For example, adapter clients can use a search expression such as "EMP%" to obtain tables starting with EMP. The adapter converts this to the following SQL query:

```
SELECT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME LIKE 'EMP%'
```

The following table lists the special characters that can be used for search and their interpretation by the SQL adapter.

Special character	Interpretation
_ (underscore)	Matches exactly one character. For example, "A_" matches "AB", "AC", "AD".
% (percentage)	Matches zero or more characters. For example, "A%" matches "A", "AB", "ABC".
[]	<ul style="list-style-type: none"> • Escapes the special meaning of _ and %. • Specifies a range or set of characters to be present. <p>For example:</p> <ul style="list-style-type: none"> • [%]% matches all names that include a % symbol. • [a-f] matches all names that have characters between and including 'a' and 'f'. • [abc] matches all names that have characters 'a', 'b', and 'c'.
[^]	Specifies a range or set of characters not to be present. For example:
	<ul style="list-style-type: none"> • [^a-f] matches all names that do not have characters between and including 'a' and 'f'. • [^abc] matches all names that do not have characters 'a', 'b', and 'c'.

◆ Important

The metadata search scope is restricted to the level immediately under the node at which the search operation is performed. For example, to search for a scalar function, you must be searching under /Scalar Function/[Schema]. Multi-level search is not supported.

Retrieving Metadata

When retrieving metadata, the SQL adapter can extract metadata under a schema, including all or a subset of database objects with the respective object and operation parameters. The adapter presents the entities from the SQL Server database as element names in XML. Because underscores are the only permissible special characters that can be included, all other special characters in the element names are encoded using underscores. For example, emp\$name is encoded as emp_x0024_name.

See Also

Other Resources

[Overview of BizTalk Adapter for SQL Server](#)

[Understanding BizTalk Adapter for SQL Server](#)

[Retrieving Metadata for SQL Server Operations in Visual Studio](#)

What Operations Can Be Performed Using the Adapter?

You can use the adapter clients to perform operations on the SQL Server database by:

- Creating BizTalk projects
- Using the WCF service model
- Using the WCF channel model

The Microsoft BizTalk Adapter for SQL Server exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel. For information about the message structure and the SOAP action associated with each operation, see [Messages and Message Schemas for BizTalk Adapter for SQL Server](#).

This section provides information about the operations supported on the SQL Server database using the SQL adapter.

In This Section

- [Performing Basic Insert, Update, Delete, and Select Operations on Tables and Views](#)
- [Operations on Tables and Views That Contain Large Data Types](#)
- [Operations on Tables and Views with User-Defined Types](#)
- [Executing Stored Procedures in SQL Server](#)
- [Executing Scalar Functions in SQL Server](#)
- [Executing Table-Valued Functions in SQL Server](#)
- [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#)
- [Operations on Tables and Views That Contain Large Data Types](#)
- [Support for Composite Operations](#)
- [Support for Polling](#)
- [Receiving Query Notifications](#)

See Also

Other Resources

[Overview of BizTalk Adapter for SQL Server](#)

Performing Basic Insert, Update, Delete, and Select Operations on Tables and Views

The Microsoft BizTalk Adapter for SQL Server surfaces a set of standard operations on each table and view in the SQL Server database. By using these operations, you can execute simple INSERT, UPDATE, SELECT, and DELETE statements qualified by a WHERE clause on the target table or view. These operations are also called data manipulation language (DML) operations.

The following table shows the DML operations that the SQL adapter supports:

Operation	Description
Insert	<p>Performs an Insert operation on the target table or view.</p> <ul style="list-style-type: none"> The Insert operation takes an array of records as input. Each record is strongly typed to the target table, and maps to the row being inserted in the table. You can insert values in identity columns provided the value of the AllowIdentityInsert binding property is set to TRUE. For more information about the AllowIdentityInsert binding property, see Working with BizTalk Adapter for SQL Server Binding Properties. The return value of the Insert operation is an array of Long data type. This array stores the identity values of the inserted rows, if any. If there is no identity column in a table, the return value is NULL. <p>Some values in the Insert operation message are treated in the following manner by the SQL adapter:</p> <ul style="list-style-type: none"> Values specified for computed columns and timestamp columns are ignored. If the node for an identity column is null, it is ignored. <p>For all other values in the Insert operation message:</p> <ul style="list-style-type: none"> If a value is specified for a column, that value is used in the INSERT statement. If the node for a particular column is null, NULL is used in the INSERT statement. <p> Note If for a particular record, there are no values that can be used in the INSERT statement (that is, either no values were specified for any column or all the column values were ignored), the adapter executes the following SQL statement: <code>insert into <table_name> default values</code></p>

Select Performs a SELECT statement on the target table or view based on an array of records (columns) and a query string that specifies a WHERE clause.

- A value must be specified for the list of the columns in the SELECT statement. If all columns have to be retrieved in a table or view, * must be specified in the SELECT statement. If specific columns have to be retrieved, the column names must be separated by comma, and specified in the same order as they are defined in the table or view.
- The WHERE clause must be included in the SELECT statement. However, if you do not want to specify a value in the WHERE clause, you can either delete the `Query` element or leave it empty.
- The Select operation also allows you to perform an Update operation. In this case, the UPDATE statement is placed inside the `Query` element of the SELECT statement.

The return value of the Select operation is a strongly-typed result set that contains the specified columns and rows from the target table or view.

Update Performs an Update operation on the target table or view.

- The Update operation takes an array of record pairs as input. Each record pair is a collection of two records, and each record is strongly-typed to the target table.
 - The first record corresponds to new values that need to be updated, that is, it corresponds to the SET clause of the UPDATE statement.
 - The second record corresponds to the old values of the rows, that is, it corresponds to the WHERE clause of the UPDATE statement.

 Note

If, for a particular record pair, there are no values that can be used in the SET clause, no UPDATE statement is executed for that record pair.

- You can update values in identity columns provided the value of the **AllowIdentityInsert** binding property is set to TRUE. For more information about the **AllowIdentityInsert** binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
- The return value of the Update operation is of Int32 data type, and denotes the number of rows updated.

Some values in the Update operation message are treated in the following manner by the SQL adapter:

- Values specified for computed columns and timestamp columns in the SET clause of the message are ignored.
- If a User Defined Type (UDT) is not byte ordered, the value specified for the UDT column in the WHERE clause is ignored.
- If the node for an identity column is null in the SET clause of the message, it is ignored.
- If the node for an identity or timestamp column is null in the WHERE clause of the message, it is ignored.
- If the node for an image, XML, Text or Ntext column is not null in the WHERE clause of the message, the values specified for them are ignored because these values cannot be compared.

For all other values in the Update operation message:

- If a value is specified for a column in the SET clause of the UPDATE statement, the value is used in the SET clause of the statement (`set <column_name> = <value>`).
- If the node for a particular column is null in the SET clause, NULL is used in the UPDATE statement (`set <column_name> = null`).
- If a value is specified for a column in the WHERE clause of the UPDATE statement, the value is used in the WHERE clause of the statement (`where <column_name> = <value>`).
- If the node for a particular column is null in the WHERE clause of the UPDATE statement, NULL is used in the UPDATE statement (`where <column_name> is null`).

Delete Performs a Delete operation on the target table or view based on a strongly-typed array of records (list of column names) of the target table and a filter string that specifies a WHERE clause.

The return value of the Delete operation is of Int32 data type, and denotes the number of rows deleted.

Some values in the Delete operation message are treated in the following manner by the SQL adapter:

- If the node for an image, XML, Text or Ntext column is not null in the WHERE clause of the message, the values specified for them are ignored because these values cannot be compared.
- If the node for an identity or timestamp column is null, it is ignored.
- If a UDT is not byte ordered, the value specified for the UDT column in the WHERE clause is ignored.

For all other values in the Delete operation message:

- If a value is specified for a column, the value is used in the WHERE clause of the DELETE statement (`where <column_name> = <value>`).
- If the node for a particular column is null, NULL is used in the DELETE statement (`where <column_name> is null`).

 **Note**

If, for a particular record, there are no values that can be used in the DELETE statement (that is, either no values were specified for any column or if all the column values present were ignored), the adapter does not execute any DELETE statement.

For more information about:

- Performing these operations using BizTalk Server, see [Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server](#).
- Message structures and SOAP action for performing DML operations, see [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Tables and Views That Contain Large Data Types

The Microsoft BizTalk Adapter for SQL Server provides supports for the following SQL Server large data types:

- Varchar(Max)
- Nvarchar(Max)
- Varbinary(Max)

To read large data values from SQL Server, the SQL adapter exposes the Select operation.

To write large data values to SQL Server, the SQL adapter exposes the Set<column_name> operation, where <column_name> is the name of the column of type Varchar(Max), Nvarchar(Max) or Varbinary(Max). The Set<column_name> operation also allows adapter clients to write FILESTREAM data in SQL Server 2008.

Note

The Set<column_name> operation is available only for those tables and views that contain columns with any of the three large data types mentioned earlier.

For detailed information about performing operations on tables and views in SQL Server that contain large data types, see [Performing Operations on Tables and Views with Large Data Types by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Operations on Tables and Views with User-Defined Types

You can use the SQL adapter to perform operations on tables or views that have columns of user-defined types (UDTs). You can use the standard table operations (Insert, Update, Delete, and Select) to read or write data into columns on UDT types. You can also execute stored procedures and functions on such tables. However, you need to perform certain tasks before you can use the adapter to operate on tables with UDT columns. Once you have performed these tasks, you can use the adapter to:

- Perform Insert, Delete, Update, and Select operations, as described in [Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server](#).
- Execute stored procedures, as described in [Executing Stored Procedures in SQL Server by Using BizTalk Server](#).
- Perform composite operations on tables with UDT columns, as described in [Performing Composite Operations on SQL Server by Using BizTalk Server](#)
- Poll tables with UDT columns, as described in [Receiving Polling-based Data-changed Messages from SQL Server by Using BizTalk Server](#).
- Perform other operations, as described in [Developing BizTalk Applications](#).

Considerations While Performing Operations on Tables with UDTs

You must perform the following tasks before you can use the adapter to perform operations on tables with UDT columns.

For SQL Server 2005

- **While generating schema for operation using Visual Studio.** Make sure the respective assemblies of the UDTs are available at the same location as the Visual Studio executable, devenv.exe. The executable is typically available at `<installation drive>:\Program Files\Microsoft Visual Studio <version>\Common7\IDE`.
- **While performing the operation using BizTalk Server.** Make sure the respective assemblies for the UDTs are available under the BizTalk Server installation location. For BizTalk Server 2006 R2, typically this is `<installation drive>:\Program Files\Microsoft BizTalk Server 2006`. For BizTalk Server 2010, typically this is `<installation drive>:\Program Files\Microsoft BizTalk Server 2010`.
- **While performing the operation using Visual Studio.** Make sure the respective assemblies for the UDTs are at the same location as the project executable file, which typically is under the project's `\bin\Debug` folder.

For SQL Server 2008

- **While generating schema for operation using Visual Studio**

UDT Type	Location of Assemblies
UDTs shipped with SQL Server 2008, for example, Geography	<ul style="list-style-type: none"> • Make sure Microsoft.SqlServer.Types.dll is added to the GAC. • Make sure SqlServerSpatial.dll is available in the System32 folder. <p>You can install these DLLs on the computer by running the SQL Server 2008 setup and selecting Management Tools – Basic and Management Tools – Complete in the Feature Selection page of the wizard.</p>
UDTs not shipped with SQL Server 2008 but defined by users	Make sure the respective assemblies for the UDTs are available at the same location as the Visual Studio executable, devenv.exe. The executable is typically available at <code><installation drive>:\Program Files\Microsoft Visual Studio <version>\Common7\IDE</code> .

- **While performing the operation using BizTalk Server**

UDT Type	Location of Assemblies
UDTs shipped with SQL Server 2008, for example, Geography	<ul style="list-style-type: none"> • Make sure Microsoft.SqlServer.Types.dll is added to the GAC. • Make sure SqlServerSpatial.dll is available in the System32 folder. <p>You can install these DLLs on the computer by running the SQL Server 2008 setup and selecting Management Tools – Basic and Management Tools – Complete in the Feature Selection page of the wizard.</p>
UDTs not shipped with SQL Server 2008 but defined by users	Make sure the respective assemblies for the UDTs are available under the BizTalk Server installation location. For BizTalk Server 2010, typically this is <installation drive>\Program Files\Microsoft BizTalk Server 2010.

• **While performing the operation using Visual Studio**

UDT Type	Location of Assemblies
UDTs shipped with SQL Server 2008, for example, Geography	<ul style="list-style-type: none"> • Make sure Microsoft.SqlServer.Types.dll is added to the GAC. • Make sure SqlServerSpatial.dll is available in the System32 folder. <p>You can install these DLLs on the computer by running the SQL Server 2008 setup and selecting Management Tools – Basic and Management Tools – Complete in the Feature Selection page of the wizard.</p>
UDTs not shipped with SQL Server 2008 but defined by users	Make sure the respective assemblies for the UDTs are available at the same location as the project executable file, which typically is under the project's \bin\Debug folder.

Once you have completed these tasks, you are all set to perform operations on tables with UDTs.

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Executing Stored Procedures in SQL Server

The Transact-SQL and CLR stored procedures in SQL Server are surfaced as operations in Microsoft BizTalk Adapter for SQL Server under the **Procedures** node while using the Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in. The operation names exposed by the SQL adapter are the same as the name of the stored procedure in SQL Server. All the parameters in the stored procedure are exposed in the corresponding operation. The OUT parameter contains the return value of the stored procedure. The result set of the stored procedure is an array of DataSet. For more information about DataSet, see <http://go.microsoft.com/fwlink/?LinkId=196853>. The schema information of the target object is obtained as part of the response message at run time.

However, if you want to obtain the schema information of the target object at design time, you must generate schemas for the procedures under the **Strongly-Typed Procedures** node in the Consume Adapter Service Add-in or Add Adapter Service Reference Plug-in. Note that the same stored procedures are surfaced under the **Procedures** and the **Strongly-Typed Procedures** node. The return value of the stored procedure is strongly typed, and not just an array of DataSet. As the schema information is available at the design time, you can use it to map the schema of the stored procedure to another schema for a different operation. For example, you can map the schema generated for a strongly-typed procedure to the schema generated for the Insert operation on a database table.

Note

You will not be able to view the schema information at design time for a strongly-typed stored procedure if:

- You are using a cursor, which is a return value of another stored procedure, as the input parameter for the strongly-typed stored procedure.
- It is a CLR stored procedure that performs some operations on a table.

Support for Stored Procedures with FOR XML Clause

The SQL adapter also enables you to execute stored procedures that have a SELECT statement with a FOR XML clause. A FOR XML clause is used in a SELECT statement to return the results as XML instead of a rowset. For more information about the FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

Note

The "native" SQL adapter available with BizTalk Server supports only those stored procedures that return XML, that is, have the FOR XML clause in the SELECT statement. With the support for stored procedures with FOR XML clause, you can execute these stored procedures using the WCF-based SQL adapter without making any changes to the stored procedure definition.

Support for Stored Procedures with Temporary Tables

The SQL adapter enables you to generate metadata for stored procedures that contain temporary tables in their definitions. However, for such stored procedures you must generate metadata by selecting the stored procedures only from the **Procedures** node, while using the Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in. The adapter does not support generating metadata for such stored procedures from under the **Strongly-Typed Procedures** node.

Support for Result Sets Containing Columns Without Names or With Same Names

The following table lists how the SQL adapter handles columns without names and same names in the result sets for stored procedures and strongly-typed stored procedures.

Result set contains...	Stored Procedure	Strongly-Typed Stored Procedure
Columns without names	The SQL adapter generates a name for the column in the following way: a unique ID (GUID) is generated for the column without "-" (hyphen), and then the GUID string is prefixed by "C" because the generated GUID might start with a digit but an XML tag name cannot.	The SQL adapter generates the following name for the column: "UnNamedColumn[column_index]", where column_index starts from '0'.

Columns with same names	The names of the columns other than the first one are appended with "_" (underscore) followed by a random GUID without "-" (hyphen). For example: "_[GUID]".	The SQL adapter does not support columns with same names in the result sets, and throws an exception. ◆ Important You must ensure that the column names in a result set have unique names.
--------------------------------	--	--

Note
In general, it is recommended that all the columns in a result set for stored procedures and strongly-typed stored procedures must be named, and have unique names.

For more information about:

- How to execute stored procedures, see [Executing Stored Procedures in SQL Server by Using BizTalk Server](#).
- How to execute stored procedures having a FOR XML clause, see [Executing Stored Procedures Having a FOR XML Clause](#).
- Message schemas for the stored procedures, see [Message Schemas for Procedures and Functions](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Executing Scalar Functions in SQL Server

The Transact-SQL and CLR scalar functions in SQL Server are surfaced as operations in Microsoft BizTalk Adapter for SQL Server. The operation name in the SQL adapter is the same as the name of the scalar function in SQL Server.

All the parameters in the scalar function are exposed in the corresponding operation. The return value of the operation in the SQL adapter is the same as the return value defined in the scalar function in SQL Server.

For more information about executing scalar functions, see [Invoking Scalar Functions in SQL Server by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Executing Table-Valued Functions in SQL Server

The Transact-SQL and CLR table valued functions in SQL Server are surfaced as operations in Microsoft BizTalk Adapter for SQL Server. The operation name in the SQL adapter is the same as the name of the table valued function in SQL Server.

All the parameters in the table valued function are exposed in the corresponding operation. If you do not specify an input parameter for a table valued function, the SQL adapter internally invokes the function using the DEFAULT keyword. This implies that the table valued function is executed using the default value specified while defining the function.

For more information about:

- Using the SQL adapter with BizTalk Server to invoke table valued functions in SQL Server, see [Invoking Table-Valued Functions in SQL Server by Using BizTalk Server](#).
- Message structure and SOAP actions for table valued functions, see [Message Schemas for Procedures and Functions](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The Microsoft BizTalk Adapter for SQL Server exposes the following operations at the root level:

- **ExecuteNonQuery:** Use this operation to execute any arbitrary SQL statements in SQL Server if you do not want any result set to be returned. You can use this operation to create database objects or change data in a database by executing UPDATE, INSERT, or DELETE statements. The return value of this operation is of Int32 data type, and:
 - For the UPDATE, INSERT, and DELETE statements, the return value is the number of rows affected by the SQL statement.
 - For all other types of statements, the return value is **-1**.
- **ExecuteReader:** Use this operation to execute any arbitrary SQL statements in SQL Server if you want the result set to be returned, if any, as an array of DataSet. For information about DataSet, see "DataSet Class" at <http://go.microsoft.com/fwlink/?LinkID=196853>.
- **ExecuteScalar:** Use this operation to execute any arbitrary SQL statements in SQL Server to return a single value. This operation returns the value only in the first column of the first row in the result set returned by the SQL statement.

Note

The advantage of ExecuteScalar over ExecuteReader is that the response message payload of the ExecuteScalar operation is much smaller compared to the one returned by the ExecuteReader operation. Therefore, if you require only one value to be returned, you should use ExecuteScalar instead of ExecuteReader.

You can use the ExecuteNonQuery, ExecuteReader or ExecuteScalar operation to execute multiple SQL statements.

For more information about performing these operations using the BizTalk Server, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Composite Operations

The Microsoft BizTalk Adapter for SQL Server enables adapter clients to perform composite operations on the SQL Server database. A composite operation can include any number of the following operations, and in any order:

- The Insert, Update, and Delete operations on the tables and views.
- Stored procedures that are surfaced as operations in the adapter.

The operations in a composite operation *must* target tables and views only in a single database.

For information about:

- How to perform composite operations in SQL adapter using the BizTalk Server, see [Performing Composite Operations on SQL Server by Using BizTalk Server](#).
- Message schemas for the composite operation, see [Message Schemas for Composite Operations](#).

◆ Important

If there are “n” number of operations in a composite operation that return a result set then “n+1” number of connections are required for the composite operation to be executed. Therefore, you must ensure that the value specified for the **MaxConnectionPoolSize** binding property is n+1 or greater. For more information about the **MaxConnectionPoolSize** binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Support for Polling

Microsoft BizTalk Adapter for SQL Server enables adapter clients to receive data-change messages from the SQL Server database. The SQL adapter supports receiving "polling-based" messages wherein the adapter executes a specified SQL statement (SELECT statement or stored procedure), retrieves or updates the data, and provides the result to the adapter client at regular intervals of time.

The SQL adapter exposes the following operations for polling:

- **Polling**: Enables you to receive periodic data-change messages for SQL Server tables or views. The messages are not strongly-typed.
- **TypedPolling**: Enables you to receive strongly-typed messages from the SQL Server database. You must use this operation if you intend to map the elements in the polling message to any other schema.
- **XmlPolling**. Enables you to use SELECT statements or stored procedures that use a FOR XML clause and return data as XML messages. This operation returns the polling message as an XML message.

For more information about FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

For more information about polling in the SQL adapter, see [Receiving Polling-based Data-changed Messages from SQL Server by Using BizTalk Server](#).

Polling

A typical polling operation using the SQL adapter involves the following:

1. The adapter clients must specify **Polling** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is **Polling**.
2. The adapter clients must specify a SQL statement for the **PolledDataAvailableStatement** binding property that determines whether there is data available for polling. The first column of the first row of the first result set returned on executing this statement contains an integer value. If there is no data available for polling, the return value is 0 (zero). If there is data available, the return value is greater than zero.
3. The adapter clients must specify a polling interval for the **PollingIntervalInSeconds** binding property to define the interval at which the statement in the **PolledDataAvailableStatement** binding property is executed. At the end of every polling interval, the polled data available statement is executed, and the result set is returned.
4. The adapter clients must specify a polling SQL statement (SELECT statement or stored procedure) for the **PollingStatement** binding property. If there is data available for polling (determined by the **PolledDataAvailableStatement** binding property), the adapter executes the polling statement to obtain and update (if applicable) the data in the SQL Server database. When the SQL adapter is used with BizTalk Server, the same transaction is also used to submit the message to BizTalk Server.
5. The adapter clients can use the **PollWhileDataFound** binding property to ignore the polling interval, and continuously poll data, as and when available.
6. The result sets that are returned as a result of executing the polling statement are sent to the adapter client as the inbound message.

Note

An XmlPolling operation involves the same steps as the Polling operation.

Strongly-Typed Polling

A typical strongly-typed polling operation using the SQL adapter involves the following:

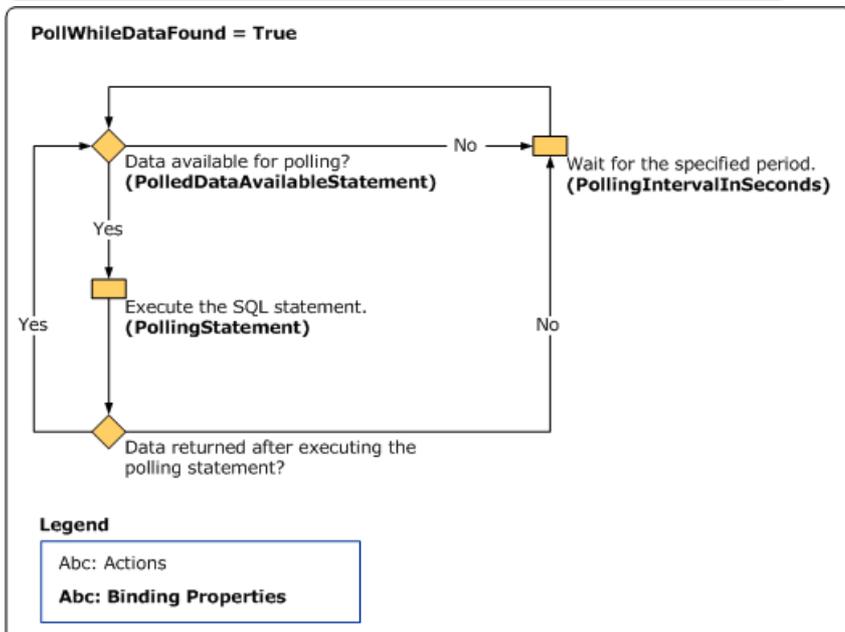
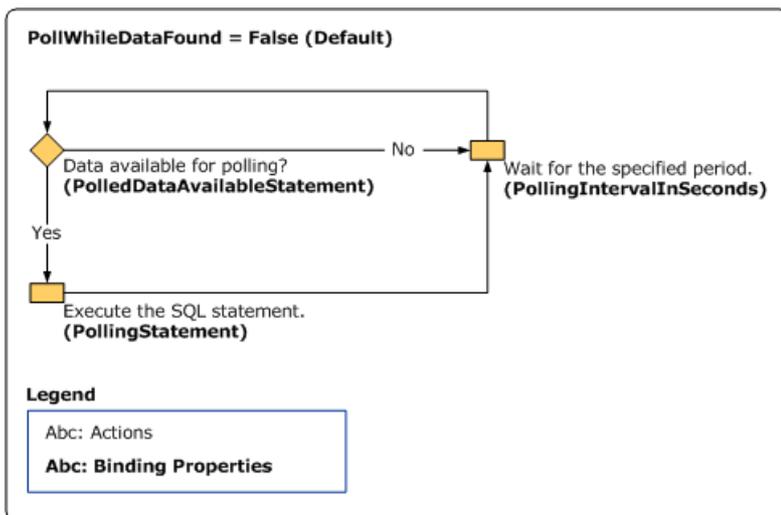
1. The adapter clients must specify **TypedPolling** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is **Polling**.
2. The adapter clients must specify an inbound ID as part of the connection URI. The inbound ID could be any string and is appended to the standard namespace of the TypedPolling operation to prevent namespace collisions.
3. The rest of the steps are same as steps 2–6 listed in the Polling operation described in the preceding section.

For detailed information about the binding properties related to polling and strongly-typed polling, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note
Multiple result sets can be returned as a result of executing the polling statement. If the result sets do not contain any rows, no messages are sent to the adapter client.

The illustration below provides information about the polling workflow in SQL adapter. Two scenarios for the polling workflow are illustrated:

1. When the value of the **PollWhileDataFound** is set to "False" (default setting).
2. When the value of the **PollWhileDataFound** is set to "True."



Though polling and query notification are both inbound operations, and inform the adapter clients about the data changes in the SQL Server database, the following table lists some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Polling	Query Notification
Polling is initiated by the adapter. The adapter executes a statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.	Query notification is initiated by SQL Server. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement.
You can use the polling statement to read or update data in a SQL Server database table.	You can use the query notification statement to only read data in a SQL Server database table.
Polling informs you about the actual data that has changed.	Query notification only informs about the type of change in the data such as Insert, Update, and Delete.
<p>The data-change notification depends on the polling interval, and the adapter clients are informed about the data changes at the end of every polling interval.</p> <p> Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last data-change notification.</p>	The data-change notification is instantaneous.

For more information about query notification in Microsoft BizTalk Adapter for SQL Server, see [Receiving Query Notifications by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Receiving Query Notifications

The adapter clients can subscribe to receive query notifications about the data changes in the SQL Server database. A SQL SELECT statement or a stored procedure specifies the data-change criteria in a table for triggering of the query notifications, and the SQL Server sends query notifications as and when the result set for the SELECT statement or the stored procedure changes.

Important

To support query notifications, the adapter clients and the SQL Server database have to fulfill certain requirements. For detailed information about these requirements, see "Enabling Query Notifications" at <http://go.microsoft.com/fwlink/?LinkID=122323>.

A typical query notification involves the following:

- The adapter clients must specify **Notification** as the inbound operation in the **InboundOperationType** binding property. The default value for this binding property is **Polling**.
- The adapter clients must specify a SQL statement to register for query notifications in the **NotificationStatement** binding property. The adapter client gets a notification from SQL Server as soon as the result set for the specified SQL statement changes.

Important

To receive notifications, the SQL statement for the notification subscription *must* meet certain criteria. For information about SQL statements that can be used for query notifications, see <http://go.microsoft.com/fwlink/?LinkId=122160>.

- The adapter clients must specify whether the adapter sends a notification to the adapter clients as soon as the listener is started in the **NotifyOnListenerStart** binding property.
- The notification is sent to the adapter clients as and when the result set of the SQL statement specified in the **NotificationStatement** binding property is changed.

For more information about these binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note

The notification subscription is always committed, regardless of whether the transaction in which the statement ran was committed or rolled back. Therefore, the notification operation might not guarantee that the result of the query subscribed for notification has changed. For example, suppose data is inserted in a table row (subscribed for notification) in a transaction, and immediately a notification is sent to the adapter informing about the change (insert). Due to some reason, the transaction rolls back, and effectively no data is inserted into the table row. However, the SQL Server does not send a notification to the adapter about the transaction roll back. For information about query notifications in SQL Server, see <http://go.microsoft.com/fwlink/?LinkId=145367>.

Differences between Query Notification and Polling

Though query notification and polling are both inbound operations, and inform the adapter clients about the data changes in the SQL Server database, the following table lists some differences between the two. The following differences will help you decide on an operation depending on your requirements:

Query Notification	Polling
Query notification is initiated by SQL Server. The notification statement issued by the adapter just instructs the database to initiate notification in case there is a change in the result set of the statement.	Polling is initiated by the adapter. The adapter executes a statement to validate whether data is available for polling, and then initiates polling by executing the polling statement if some data is available for polling.

<p>You can use the query notification statement to only read data in a SQL Server database table.</p>	<p>You can use the polling statement to read or update data in a SQL Server data base table.</p>
<p>Query notification informs only about the type of change in the data such as Insert, Update, and Delete.</p>	<p>Polling informs you about the actual data that has changed.</p>
<p>The data-change notification is instantaneous.</p>	<p>The data-change notification depends on the polling interval, and the adapter clients are informed about the data changes at the end of every polling interval.</p> <p> Tip Polling can give you better throughput in scenarios where the data changes are happening continuously, and you do not want to be notified of each change as and when it happens. Instead, you specify a polling interval after which you want to be notified of all the changes that have happened since the last change notification.</p>

For more information about query notification in Microsoft BizTalk Adapter for SQL Server, see [Receiving Query Notifications by Using BizTalk Server](#).

See Also

Other Resources

[What Operations Can Be Performed Using the Adapter?](#)

Other Features Supported by the Adapter

In addition to the features discussed throughout the topics of [Overview of BizTalk Adapter for SQL Server](#), the Microsoft BizTalk Adapter for SQL Server also provides the following features that are useful for adapter clients:

- **Support for configuring adapters using binding properties.** Adapter clients can configure the SQL adapter by specifying certain binding properties. For example, clients can specify the maximum number of connections allowed in a connection pool for a specific connection string by setting the **MaxConnectionPoolSize** binding property. For more information, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
- **Support for null values for operation parameters.** Adapter clients can provide null values for operation parameters using the XSD "nillable" attribute.
- **Support for dynamic ports in BizTalk.** Through the BizTalk WCF-Custom adapter, the SQL adapter supports a dynamic port that enables dynamic routing of messages from BizTalk Server based on the message context properties. For more information, see [Configuring Dynamic Ports](#).
- **Support for performance counters.** The SQL adapter supports WCF-based performance counters for use by adapter clients. For more information about performance counters, see [Using Performance Counters](#).

See Also

Other Resources

[Overview of BizTalk Adapter for SQL Server](#)

New Features in BizTalk Adapter for SQL Server

This section lists the new features in Microsoft BizTalk Adapter for SQL Server.

New Features in the SQL Adapter

Following are the new features introduced in this release of the SQL adapter.

Technology-Related Features

Feature	Comment
Use of Windows Communication Foundation (WCF)	The SQL adapter is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK (WCF LOB Adapter SDK). In turn, the WCF LOB Adapter SDK is built on top of WCF. The adapter is exposed as a WCF channel to adapter clients. This enables connectivity, metadata exchange, and business data exchange with external systems.
Support for the WCF channel model and the WCF service model	In the WCF channel model, adapter clients can consume the SQL adapter by directly sending and receiving XML messages. In the WCF service model, adapter clients can generate a .NET proxy class from the Web Services Description Language (WSDL) obtained by using the SQL adapter.
Support for 64-bit platforms	The SQL adapter is now available for 64-bit platforms.

Metadata-Related Features

Feature	Comment
Browse, search, and retrieve metadata	The adapter clients can browse and search metadata in batches by specifying a batch size. This feature is available only when programming into the adapter and not through the Consume Adapter Service BizTalk Project Add-in. The metadata search is supported at the Tables, Views, Procedures, Scalar Functions, and Table Valued Functions levels. The search string is used directly within a SQL statement.
Support for invoking artifacts with same name in different databases	In the SQL adapter, the namespaces in the XML Schema Definition (XSD) file contained only the schema name, and in some cases the object name. However, if an application wants to execute operations on identically named artifacts with different metadata in different databases, the generated metadata will conflict. The only way to distinguish the metadata is to use the database name in the XSD namespaces. The current version of the SQL adapter allows you to specify the database name in the XSD namespaces by setting the value of the UseDatabaseNameInXsdNamespace binding property to TRUE. The default value of the binding property is false, which implies that the XSD namespaces will not contain the database name. For more information about the UseDatabaseNameInXsdNamespace binding property, see Working with BizTalk Adapter for SQL Server Binding Properties .

Performance-Related Features

Feature	Comment
Support for performance counters	The SQL adapter supports WCF-based performance counters for use by adapter clients. For more information about performance counters, see Using Performance Counters .

Operations-Related Features

Feature	Comment
Support for the new SQL Server 2005 and SQL Server 2008 data types	The SQL adapter supports the following new data types introduced in: <ul style="list-style-type: none"> SQL Server 2005: XML, Varchar(Max), and Varbinary(Max). SQL Server 2008: Date, Time, Datetimeoffset, Datetime2, Hierarchyid, Geography, Geometry, and FILESTREAM.

Support for User-Defined Types (UDTs)	The SQL adapter supports performing operations on tables and views that contain UDTs. For information about support for UDTs, see Operations on Tables and Views with User-Defined Types .
Support for executing Transact-SQL and CLR stored procedures and functions	<p>Adapter clients can execute Transact-SQL and CLR:</p> <ul style="list-style-type: none"> • Stored procedures in a SQL Server database. • Scalar and table valued functions in a SQL Server database. <p>For more information about this, see What Operations Can Be Performed Using the Adapter?.</p>
Support for executing stored procedures with or without the FOR XML clause	The SQL adapter enables you to execute stored procedures that have a SELECT statement with or without a FOR XML clause. The previous version of the adapter supported only those stored procedures that had a FOR XML clause in the SELECT statement. For information about executing stored procedures, see Executing Stored Procedures in SQL Server .
Support for streaming of large objects	<p>Adapter clients can stream large character and binary fields in the SQL Server database using the Set<column name> operation, where <column_name> is the name of the column of type Varchar(Max), Nvarchar(Max) or Varbinary(Max). The Set<column name> operation also allows you to insert or update FILESTREAM data in a SQL Server 2008 database. For more information about this, see Operations on Tables and Views That Contain Large Data Types.</p> <p> Note To read character and binary fields in SQL Server tables and views, adapter clients use the Select operation .</p>
Support for query notifications	Adapter clients can receive query notifications from SQL Server based on a triggering SELECT statement or a stored procedure. The notification is sent by the SQL Server to the adapter clients as and when the result set for the SELECT statement or the stored procedure changes. For more information about query notifications, see Receiving Query Notifications .
Support for executing arbitrary SQL statements	The SQL adapter enables adapter clients to execute arbitrary SQL statements using the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations. For more information about these operations, see Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations .
Support for composite operations	<p>The SQL adapter enables adapter clients to perform composite operations on the SQL Server database. A composite operation can include any number of the following operations, and in any order:</p> <ul style="list-style-type: none"> • The Insert, Update, and Delete operations on the tables and views. • Stored procedures that are surfaced as operations in the adapter. <p>For more information about composite operations, see Support for Composite Operations.</p>
Enhanced polling	The SQL adapter now supports two additional types of polling: TypedPolling and XmlPolling . For information about these polling types, see Support for Polling .
Support for performing operations on artifacts in multiple schemas	Apart from the default schema (dbo), adapter clients can also perform operations on artifacts in other schemas in the SQL Server database provided that the user credentials used to connect using the SQL adapter has access to those schemas in the SQL Server database. For information about a schema in SQL Server database, see http://go.microsoft.com/fwlink/?LinkId=130148 .

See Also

Other Resources

[Overview of BizTalk Adapter for SQL Server](#)

Limitations of BizTalk Adapter for SQL Server

The following are known limitations for Microsoft BizTalk Adapter for SQL Server:

- The SQL adapter does not support synonyms created in the SQL Server database. For information about synonyms in SQL Server, see <http://go.microsoft.com/fwlink/?LinkId=120111>.
- If you change the system time of the computer running the BizTalk Server host, the time is not updated automatically in the BizTalk Server host. This could lead to incorrect behavior of the inbound operations that use the receive port of BizTalk Server. As a workaround, you must restart the host instance that has a receive port after you have changed the system time of the computer running it.
- If a parameter name in a stored procedure contains 127 or more characters, you cannot execute the stored procedure using the SQL adapter. This is due to the limitation of ADO.NET.
- The WSDL the SQL adapter generates, when converted to a proxy, exposes the DateTimeOffset column as System.DateTime. This data type cannot store time zone information. As a consequence, any date/time value the adapter sends to the proxy will be converted into local time in the .NET application. If you wish to keep the time zone information, you must change the interface of your proxy to use the String type instead of System.DateTime. Then, use XmlConvert.ToDateTimeOffset to create a System.DateTimeOffset object, which can store the timezone information.

See Also

Other Resources

[Understanding BizTalk Adapter for SQL Server](#)

SQL Adapter Tutorials

The first of the SQL adapter tutorials contains detailed instructions on how to migrate BizTalk projects created using the previous version of the SQL adapter to use the WCF-based SQL adapter. The other SQL adapter contains, scenario-based, step-by-step instructions on performing operations on SQL Server using the SQL adapter.

Also, the topics in the [Developing BizTalk Applications](#) section provide detailed instructions on how to use the SQL adapter with BizTalk Server.

In This Section

- [Tutorial 1: Migrating BizTalk Projects](#)
- [Tutorial 2: Employee - Purchase Order Process](#)

See Also

Other Resources

[Getting Started](#)

Tutorial 1: Migrating BizTalk Projects

The previous version of the SQL adapter that shipped with Microsoft BizTalk Server differs from the WCF-based SQL adapter in many aspects, including:

- The design-time experience of creating a BizTalk project.
- The metadata retrieval experience.
- Schema file name and namespace.
- Data type mappings.
- The operations that can be performed using the adapter.
- Physical port configuration in the BizTalk Server Administration console

These differences are explained in the topics within Migrating BizTalk Projects Created Using the Previous Version of the SQLAdapter.

However, you can make changes to the BizTalk project that was created using the previous version of the adapter and make it work with the WCF-based SQL adapter.

This tutorial provides instructions on the changes you should make to the existing BizTalk project created using the previous version of the adapter.

Note

In this tutorial, for the sake of brevity, the previous version of the SQL adapter will be referred to as vPrev SQL adapter. Similarly, a BizTalk project that uses the vPrev SQL adapter will be referred to as vPrev BizTalk project.

Important

This tutorial provides guidance on how to migrate a vPrev SQL adapter BizTalk project that performs a basic insert operation on a SQL Server database table. This tutorial does not cover all possible scenarios for migration from the vPrev SQL adapter to the new WCF-based SQL adapter. You must use this migration tutorial as a foundation and modify accordingly to make changes that are relevant to your existing project.

Sample Used for the Tutorial

This tutorial is based upon a sample (SQL_Migration) that demonstrates how to migrate a vPrev BizTalk project. The sample is provided with Microsoft BizTalk Adapter Pack. For more information, see Samples.

Prerequisites

- You must have a vPrev BizTalk project. This tutorial involves a BizTalk project that performs an Insert operation on a Customer table in the SQL Server database. The Customer table has the following design:

Column Name	Description
v_custid	Primary key, integer type, identity field
Name	nchar(10) type

- You must have a request message to perform an Insert operation on the SQL Server database using the vPrev SQL adapter. The request message must conform to the schema of the Insert operation generated using the vPrev SQL adapter.
- You should have completed the steps in [Before You Develop BizTalk Applications](#).

Understanding a BizTalk Project Created Using the Previous Version of the Adapter

The key constituents of a vPrev BizTalk project created are:

- **BizTalk orchestration.** This is a simple orchestration that picks request messages from a file location, sends the request message to the SQL Server database using a WCF-Custom send-receive port, receives the response, and saves it to another file location.
- **Schema for the operation you wish to perform on the SQL Server database.** This tutorial involves a BizTalk project that performs an Insert operation on the Customer table. The schema generated for the Customer table is InsertCustomerService.xsd. This schema is generated using the vPrev SQL adapter.
- **Request message.** The request message to perform an Insert operation on the Customer table. The schema of the request message conforms to the schema of the Insert operation as surfaced by the previous version of the SQL adapter.

How to Migrate a BizTalk Project Created Using the Previous Version of the Adapter

The goal of this migration tutorial is to enable you to send a request message, which conforms to schema generated by vPrev SQL adapter, using a WCF-Custom port that can only process messages conforming to the WCF-based SQL adapter. So, in short, the migration exercise involves configuring the WCF-Custom port to process messages that do not conform to the WCF-based SQL adapter's schema.

However, to be able to configure the WCF-Custom port appropriately, you must perform the following tasks:

- Generate metadata for the Insert operation on the Customer table using the WCF-based SQL adapter.
- Map the request message for performing an Insert operation using the vPrev SQL adapter to a request message for performing an Insert operation using the WCF-based SQL adapter.
- Map the response message received using the WCF-based SQL adapter to the response message for the vPrev SQL adapter.
- Create a WCF-Custom SQL send-receive port in the BizTalk Server Administration console.
- Configure the WCF-Custom port to use the request and response mappings.

In This Section

- [Step 1: Modify the vPrev BizTalk Project](#)
- [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#)
- [Step 3: Test the Migrated Application](#)

See Also

Other Resources

[SQL Adapter Tutorials](#)

Step 1: Modify the vPrev BizTalk Project



Step 1

Time to complete: 10 minutes

Objective: In this step, you make the following changes to the existing vPrev BizTalk project:

- Generate metadata for the Insert operation on the Customer table using the WCF-based SQL adapter.
- Map the request message for performing an Insert operation using the vPrev SQL adapter to a request message for performing an Insert operation using the WCF-based SQL adapter.
- Map the response message received using the WCF-based SQL adapter to the response message received using the vPrev SQL adapter.

Prerequisites

You must have a vPrev BizTalk project to perform an Insert operation on the Customer table in the SQL Server database.

To modify the vPrev BizTalk project

1. Generate metadata for the Insert operation on the Customer table using the WCF-based SQL adapter. You can use the Consume Adapter Service BizTalk Project Add-in to generate metadata.

For instructions on how to generate metadata, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#). After the schema is generated, a file with the name similar to *TableOperation.dbo.Customer.xsd* is added to the BizTalk project. This file contains the schema for sending a message to perform an Insert operation on the Customer table in the SQL Server database using the WCF-based SQL adapter.

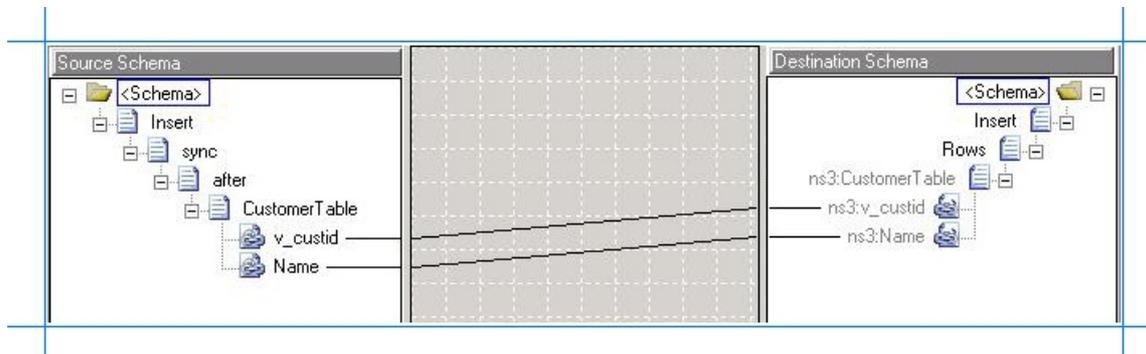
2. Generating the metadata for the Insert operation also creates a port binding file. In the next step, this binding file will be used to create a WCF-Custom send port to send messages to the SQL Server database. The SOAP action for the operation is also set to the operation for which you generated metadata. For example, if you generate metadata for the Insert operation, the operation name in the SOAP action on the send port will be "Insert". However, the operation name on the logical send port that you create as part of the orchestration could be different, for example, "Operation_1". As a result, when you send messages to the SQL Server database using the send port, you get an error. To prevent this, make sure the operation name on the logical send port in your orchestration is the same as the operation name for which you generated metadata.

So, in case of this tutorial, because you generate metadata for the Insert operation, change the name of the logical send port operation to "Insert".

3. For the request message, map the schema generated using vPrev SQL database adapter to the schema generated using the WCF-based SQL adapter.
 - a. Add a BizTalk Mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and then click **New Item**.

In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **RequestMap.btm**. Click **Add**.
 - b. From the Source Schema pane, click **Open Source Schema**.
 - c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and then select the schema for the request message for the vPrev SQL adapter. For this tutorial, select *New_Migration.InsertCustomerService*, and then click **OK**.
 - d. In the **Root Node for Source Schema** dialog box, select *Insert*, and then click **OK**.
 - e. From the Destination Schema pane, click **Open Destination Schema**.

- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and then select the schema for the request message for the WCF-based SQL adapter. For this tutorial, select *New_Migration.TableOperation.dbo.Customer*, and then click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *Insert*, and then click **OK**.
- h. Map the respective elements in both the schemas as illustrated in the following figure.



- i. Save the map.

4. For the response message, map the schema generated using vPrev SQL adapter to the schema generated using the WCF-based SQL adapter.

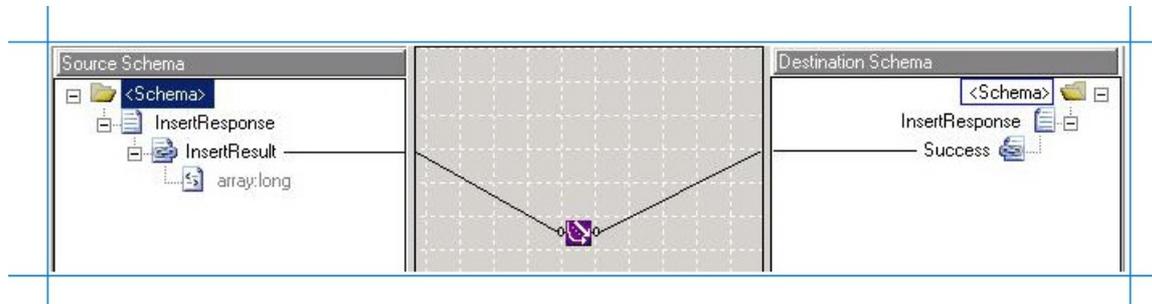
- a. Add a BizTalk Mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.

In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **ResponseMap.btm**, and then click **Add**.

- b. From the Source Schema pane, click **Open Source Schema**.
- c. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and then select the schema for the response message for the WCF-based SQL adapter. For this tutorial, select *New_Migration.TableOperation.dbo.Customer*, and then click **OK**.
- d. In the **Root Node for Source Schema** dialog box, select *InsertResponse*, and then click **OK**.
- e. From the Destination Schema pane, click **Open Destination Schema**.
- f. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and then select the schema for the response message for the vPrev SQL adapter. For this tutorial, select *New_Migration.InsertCustomerService*, and then click **OK**.
- g. In the **Root Node for Target Schema** dialog box, select *InsertResponse*, and then click **OK**.
- h. You will notice a few differences between the response schemas for generated by the two adapters. These differences can be explained as follows:
 - Using the WCF-based SQL adapter, if you insert a record into a table that contains a primary key (and is also an identity field) the response for the Insert operation returns the value for the identity field for the inserted row. So, the schema for the response message conforming to the WCF-based SQL adapter contains an additional *InsertResult* element. This element contains an array, which in turn contains the identity fields for the inserted rows.
 - Using the vPrev SQL adapter, if you insert a record into a table, the adapter only returns an empty "Success" element as part of the response message.

So, the schemas are mapped in such a way that the response from the WCF-based SQL adapter containing the value for the identity fields is “copied” as part of the “Success” element, which is part of the response message from the vPrev SQL adapter. You can use the Mass Copy functoid to copy elements from one schema into another.

To use the Mass Copy functoid, from the **Toolbox**, drag the Mass Copy functoid and drop it on the Mapper grid. Connect the **InsertResult** element in the source schema to the functoid. Similarly, connect the **Success** element in the destination schema to the functoid. The following figure illustrates how the two elements are mapped via the functoid.



Note
For more information about the Mass Copy functoid, see <http://go.microsoft.com/fwlink/?LinkId=119749>.

i. Save the map.

5. Save and build the BizTalk solution. Right-click the solution, and then click **Build Solution**.
6. Deploy the solution. Right-click the solution, and then click **Deploy Solution**.

Next Steps

Create a WCF-custom send port, and configure it to use the maps you created in this step, as described in [Step 2: Configure the Orchestration in BizTalk Server Administration Console](#).

See Also

Concepts

[Tutorial 1: Migrating BizTalk Projects](#)

Step 2: Configure the Orchestration in BizTalk Server Administration Console



Time to complete: 10 minutes

Objective: In this step, you perform the following tasks:

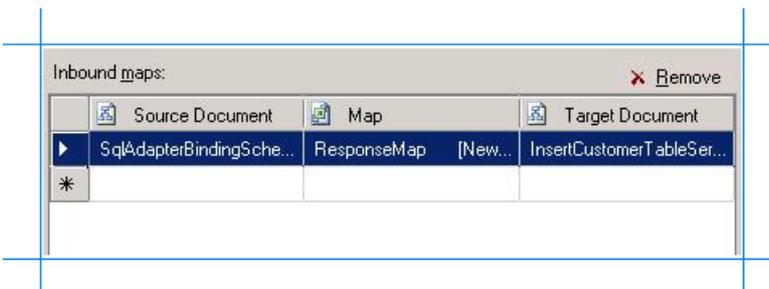
- Create a WCF-Custom send-receive port to send and receive messages from the SQL Server database using the SQL adapter. Configure this port to use the maps you created in the previous step.
- Configure the orchestration you deployed in the previous step to use the WCF-Custom port.

Prerequisites

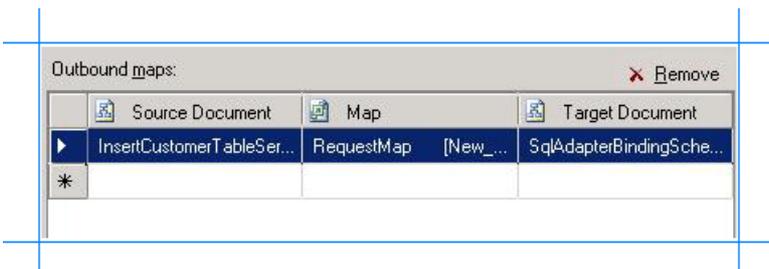
You should have deployed the BizTalk orchestration for which you want to configure the WCF-Custom port as described in [Step 1: Modify the vPrev BizTalk Project](#).

To create a WCF-Custom port

1. When you generate schema for an operation on the SQL Server database using Consume Adapter Service BizTalk Project Add-in, a binding file is also added to the BizTalk project. You can import this binding file into your BizTalk application to create a WCF-Custom send-receive port. For instructions on importing a binding file, see [Importing Bindings](#).
2. After you import the binding file, a send port is created under the **Send Ports** folder in the BizTalk Server Administration console.
3. Right-click the WCF-Custom port, and then click **Properties**.
4. From the left pane of the send port properties dialog box, click the **General** tab. From the right pane, click **Configure**.
5. In the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, specify the credentials to connect to a SQL Server database, and then click **OK**.
6. From the left pane of the send port properties dialog box, click **Inbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **ResponseMap**.



7. From the left pane of the send port properties dialog box, click **Outbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **RequestMap**.



8. Click **OK**.

To configure the BizTalk application

1. In the BizTalk Server Administration console, expand **BizTalk Group**, expand **Applications**, and then expand the BizTalk Application where the orchestration is deployed.
2. Right-click the BizTalk application, and then select **Configure**.
3. From the left pane, click the orchestration to configure. From the right pane, from the **Host** drop-down list, select a BizTalk host instance.
4. Under the **Bindings** box, map the logical ports of the BizTalk orchestration to the physical ports in the BizTalk Server Administration console.
 - a. Select the file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
 - b. Select the file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
 - c. Select the WCF-Custom send port you created earlier in this topic.
 - d. Click **OK**.

Next Steps

You have now completed migration of your vPrev BizTalk project to a BizTalk project that sends messages to the SQL Server database using the WCF-based SQL adapter. You must now test the migrated BizTalk application by sending a request message to perform an Insert operation on the SQL Server database, as described in [Step 3: Test the Migrated Application](#).

See Also

Concepts

[Tutorial 1: Migrating BizTalk Projects](#)

Step 3: Test the Migrated Application



Time to complete: 5 minutes

Objective: In this step, you will test the migrated application by performing an Insert operation on the Customer table. To do this, you drop a request message that conforms to the schema generated using the vPrev SQL adapter.

Prerequisites

- Configure the BizTalk application by mapping the logical ports in the BizTalk orchestration to physical ports in the BizTalk Server Administration console.
- Configure the BizTalk application to use the WCF-Custom send port for the WCF-based SQL adapter.

To test the migrated application

1. Create a request XML that conforms to the schema generated by the vPrev SQL adapter. Using the outbound map, the WCF-Custom send port converts this to conform to the schema for the WCF-based SQL adapter and sends it to the SQL Server database.

```
<Insert xmlns="http://SQLInsert">
  <sync>
    <after>
      <CustomerTable Name="John" />
    </after>
  </sync>
</Insert>
```

2. Paste the request message to the folder that is mapped to the file receive location.
3. The orchestration consumes the request message and sends it to the SQL Server database. The response from the SQL Server database is received in the schema that conforms to the schema of the WCF-based SQL adapter. Using the inbound map, the WCF-Custom send port converts this to the schema for the vPrev SQL adapter. The response from the SQL Server database is saved to the other file location defined as part of the orchestration. The response for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<InsertResponse xmlns="http://SQLInsert">
  <Success>
    <long xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">101</long>
  </Success>
</InsertResponse>
```

In the preceding response, "101" is the value of the identity column inserted in the Customer table.

See Also

Concepts

[Tutorial 1: Migrating BizTalk Projects](#)

Tutorial 2: Employee - Purchase Order Process

In this tutorial, you are automating the process where the Purchases department that places an equipment order every time a new employee joins the organization. Both employee details and purchase order details are maintained in **Employee** and **Purchase_Order** tables respectively, in a SQL Server database. The Purchases department is informed by updating the **Purchase_Order** table in the SQL Server database and by sending an e-mail. Within the process, the following actions occur:

1. The adapter receives a notification each time the **Employee** table is updated. The adapter then sends a notification to the BizTalk orchestration.
2. The BizTalk orchestration figures out whether the notification is for a new record inserted into the **Employee** table. If the notification is for any other operation on the **Employee** table, the orchestration does not perform any operation.
3. If the notification is for an Insert operation on the **Employee** table, notifying that a new employee record was added, the orchestration uses the SQL adapter to read the details of the new record.
4. The orchestration receives a response that contains the details of the new added employee record. The orchestration maps the **Employee_ID** and **Designation** fields in the response to the request message for the Insert operation on the **Purchase_Order** table.
5. The orchestration then uses the SQL adapter to perform an Insert operation on the **Purchase_Order** table. The response for the Insert operation is sent to the Purchases department as an e-mail.

About the Database Objects Used in this Sample

This tutorial uses the database objects created by the SQL script shipped with the samples. For more information about the script and the samples, see [Samples](#). The database objects that you will use in this tutorial are:

- **ADAPTER_SAMPLES** database.
- **Employee** and **Purchase_Order** tables.
- **UPDATE_EMPLOYEE** stored procedure.

All these database objects are created when you run the SQL script provided with the sample. Make sure you run the script before you start with the tutorial.

Sample Based on This Tutorial

A sample, **Employee_PurchaseOrder**, which is based on this tutorial is also provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

We recommend that you go through the tutorial completely to understand how to create BizTalk projects using the adapter, and then look at the sample as a reference.

In This Section

- [Lesson 1: Generate Schemas and Create Messages](#)
- [Lesson 2: Receive and Filter Notifications](#)
- [Lesson 3: Execute a Stored Procedure to Select New Employees Added](#)
- [Lesson 4: Perform an Insert Operation on the Purchase Order Table](#)
- [Lesson 5: Deploy the Solution](#)

Lesson 1: Generate Schemas and Create Messages

In this lesson, you create a BizTalk project in Visual Studio, generate the schemas for the operations you perform on the SQL Server database using the SQL adapter, and then create messages for those schemas.

In This Section

- [Step 1: Generate Schema for Operations](#)
- [Step 2: Create Messages for BizTalk Orchestrations](#)

Step 1: Generate Schema for Operations



Step 1

Time to complete: 5 minutes

Objective: In this step, you generate schemas for the operations that you perform on the SQL Server database using the SQL adapter. For this tutorial, you must generate schema for the following:

- **Notification** (inbound operation).
- **UPDATE_EMPLOYEE** stored procedure (outbound operation).
- **Insert** operation on the **Purchase_Order** table (outbound operation).

Prerequisites

Before you proceed with the tutorial, make sure:

- You must have completed the steps in [Before You Develop BizTalk Applications](#).
- You must log on as a member of the BizTalk Server Administrators group.

To generate schema for operations

1. Create a new BizTalk project in Visual Studio. For this tutorial, name the project as **Employee_PurchaseOrder**.
2. Connect to the ADAPTER_SAMPLES SQL Server database using the Consume Adapter Service Add-in. For instructions on how to connect using Consume Adapter Service Add-in, see [Connecting to SQL Server in Visual Studio Using Consume Adapter Service Add-in](#).

Note

You can also connect to SQL Server using the Add Adapter Metadata Wizard. However, for this tutorial you will use the Consume Adapter Service Add-in.

3. Generate schema for the **Notification** inbound operation.
 - a. After connecting to the ADAPTER_SAMPLES database, in the Consume Adapter Service Add-in, from the **Select contract type** list, select **Service (Inbound operations)**.
 - b. From the **Select a category** box, click the root node (/).
 - c. From the **Available categories and operations** box, select **Notification** and click **Add**. The **Notification** operation is now displayed in the **Added categories and operations** box. Click **OK**.
4. Generate schema for the **UPDATE_EMPLOYEE** stored procedure and the Insert operation on **Purchase_Order** table.
 - a. Repeat step 2 to connect to ADAPTER_SAMPLES database in SQL Server using the Consume Adapter Service Add-in.

Note

You cannot generate schema for inbound and outbound operations at the same time. Hence, in step 3, after you click **OK** to generate the schema for **Notification** operation, the Consume Adapter Service Add-in closes. You must reconnect to the SQL Server database to generate schema for outbound operations.

- b. From the **Select contract type** list, select **Client (Outbound operations)**.

- c. From the **Select a category** box, click the **Strongly-Typed Procedures** node. From the **Available categories and operations** box, select **UPDATE_EMPLOYEE**, and then click **Add**.

◆ **Important**

The **UPDATE_EMPLOYEE** stored procedure is also available under the **Procedures** node. However, if you generate the schema for the stored procedure from under the **Procedures** node, the response message schema is not available at design-time but is received with the response message after you execute the stored procedure.

In this tutorial, you will map the response schema of the stored procedure to the input schema of the Insert operation on the **Purchase_Order** table. Therefore, you will need the schema for the **UPDATE_EMPLOYEE** stored procedure at design-time and you must select the stored procedure from under the **Strongly-Typed Procedures**. By doing so, you will get the schema of the stored procedure at design-time.

- d. From the **Select a category** box, expand the **Tables** node, and click the node for **Purchase_Order** table. From the **Available categories and operations** box, select **Insert**, click **Add**, and then click **OK**.

What did I just do?

In this step, you generated schemas for **Notification** (inbound operation), **UPDATE_EMPLOYEE** stored procedure, and **Insert** operation on the **Purchase_Order** table. After you generate the schema, the Consume Adapter Service Add-in adds the following files to your BizTalk project:

- XSD files that contain schema for the request message to invoke operations on SQL Server.
- XML binding files that you can use to create WCF-Custom send and receive ports in BizTalk Server Administration console.

For more information about generating schemas, see [Browsing, Searching, and Retrieving Metadata for SQL Server Operations](#).

Next Steps

You create messages in the BizTalk project for the schemas in [Step 2: Create Messages for BizTalk Orchestrations](#).

See Also

Other Resources

[Lesson 1: Generate Schemas and Create Messages](#)

Step 2: Create Messages for BizTalk Orchestrations



Time to complete: 5 minutes

Objective: In this step, you add an orchestration to the BizTalk project and create messages for the schemas you generated in [Step 1: Generate Schema for Operations](#).

Prerequisites

You must have completed [Step 1: Generate Schema for Operations](#).

To create messages in an orchestration

1. Add a BizTalk orchestration to the BizTalk project in Visual Studio:
 - a. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**.
 - b. In the **Add New Item** dialog box, from the **Categories** box, click **Orchestration Files**. From the **Templates** box, click **BizTalk Orchestration**.
 - c. Type a name for the BizTalk orchestration, and then click **Add**. For this tutorial, enter the name **EmployeeOrch.odx**.
2. Open the **Orchestration View** window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. Add messages to the orchestration.
 - a. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
 - b. Right-click the newly created message, and then select **Properties Window**.
 - c. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select Employee_PurchaseOrder.Notification , where Employee_PurchaseOrder is the name of your BizTalk project. Notification is the schema generated for the Notification operation.

- d. Repeat the previous step to add four new messages—a request-response message set for invoking the UPDATE_EMPLOYEE stored procedure and another request-response message set for performing the **Insert** operation on **Purchase_Order** table.

Set Identifier to	Set Message Type to
UpdateEmployee	<i>Employee_PurchaseOrder.TypedProcedure_dbo.UPDATE_EMPLOYEE</i> , where TypedProcedure_dbo.UPDATE_EMPLOYEE is the schema for the UPDATE_EMPLOYEE stored procedure.
UpdateEmployeeResponse	<i>Employee_PurchaseOrder.TypedProcedure_dbo.UPDATE_EMPLOYEEResponse</i>

InsertPO	<i>Employee_PurchaseOrder.TableOperation_dbo_Purchase_Order.Insert</i> , where TableOperation_dbo_Purchase_Order.Insert is the schema for the Insert operation on the Purchase_Order table.
InsertPOResponse	<i>Employee_PurchaseOrder.TableOperation_dbo_Purchase_Order.InsertResponse</i>

- e. Save the orchestration file and the BizTalk project.

What did I just do?

In this step, you created messages for invoking performing inbound and outbound operations on SQL Server using the SQL adapter.

Next Steps

You add orchestration shapes to receive notification from SQL Server and filter notifications for Insert operation, as described in [Lesson 2: Receive and Filter Notifications](#).

See Also

Tasks

[Step 1: Generate Schema for Operations](#)

Other Resources

[Lesson 1: Generate Schemas and Create Messages](#)

Lesson 2: Receive and Filter Notifications

In this lesson, you start creating an orchestration that receives notifications for changes to the **Employee** table. After the orchestration receives the notification, it extracts the type of notification and if the notification type is for an Insert operation on the **Employee** table, an "if" condition is used to perform subsequent tasks. In this lesson, you will perform the following tasks:

1. Add a one-way receive port and a **Receive** shape to the orchestration to receive the notification message.
2. Add an **Expression** shape that contains an xpath query to extract the type of notification.
3. After the notification type is known, add a filter to the orchestration by including a **Decide** shape. This shape decides whether the notification is for an Insert notification and then performs subsequent operations. If the notification is not for an Insert operation, the orchestration does not do anything.

In This Section

- [Step 1: Add Orchestration Shapes to Receive Notification](#)
- [Step 2: Extract Notification Type from Notification Message](#)
- [Step 3: Add a Filter for Insert Notifications](#)

Step 1: Add Orchestration Shapes to Receive Notification



Step 1

Time to complete: 5 minutes

Objective: In this step, you add orchestration shapes to receive notification for changes to the **Employee** table.

Prerequisites

You must have completed the steps in [Lesson 1: Generate Schemas and Create Messages](#).

To receive notification messages

1. Open the BizTalk orchestration, **EmployeeOrch.odx**, you added in [Step 2: Create Messages for BizTalk Orchestrations](#).
2. Add a **Receive** shape to the orchestration. From the orchestration Toolbox, drag the **Receive** shape to the orchestration design surface, and drop it into the space indicated between the **Begin** (green circle) and **End** (red octagon) shapes.

Set this property	To this value
Activate	True
Message	NotifyReceive
Name	ReceiveNotification

3. Add a one-way receive port to the orchestration. You will use this port to receive notification messages from the SQL Server database. Set the following properties for the port.

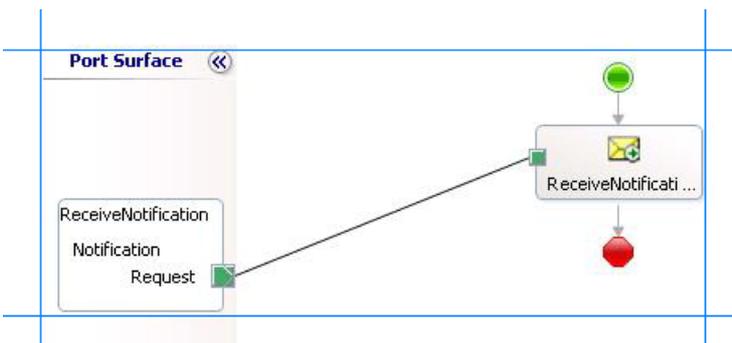
Set this property	To this value
Communication Direction	Receive
Communication Pattern	One-Way
Identifier	ReceiveNotification

Also, change the operation name from Operation_1 to **Notification**.

4. Connect the **ReceiveNotification** port to the **ReceiveNotification** action shape. In Orchestration Designer, on the design surface, drag the green arrow-shaped handle for the port to the corresponding green handle of the action shape.

Note
 In this step, you use the drag-and-drop method to connect ports to action shapes. You could instead use the operation property of an action shape to connect the action shape to a port.

5. The following figure shows the in-progress orchestration.



What did I just do?

In this step, you added orchestration shapes and receive port to receive notification from the SQL Server database.

Next Steps

You add an expression shape to the orchestration to extract the type of notification received from the SQL Server database, as described in [Step 2: Extract Notification Type from Notification Message](#).

See Also

Tasks

[Step 2: Extract Notification Type from Notification Message](#)

Other Resources

[Lesson 2: Receive and Filter Notifications](#)

Step 2: Extract Notification Type from Notification Message



Time to complete: 5 minutes

Objective: In this step, you add an expression shape to extract the type of notification received from the SQL Server database.

Prerequisites

You must have completed [Step 1: Add Orchestration Shapes to Receive Notification](#).

To extract the notification type from the notification message

1. Add a variable to the BizTalk orchestration you created in [Step 1: Add Orchestration Shapes to Receive Notification](#).
 - a. From the Orchestration View, right-click **Variables**, and then click **New Variable**.
 - b. Right-click the new variable, **Variable_1**, and click **Properties Window**. Set the following properties for the variable.

Set this property	To this value
Identifier	NotificationType
Type	System.String

2. Add an **Expression** shape to the BizTalk orchestration. From the orchestration Toolbox, drag the **Expression** shape to the orchestration design surface, and drop it after the **Receive** shape

Within the **Expression** shape, you will add an xpath query to extract the type of notification message received from SQL Server. Before creating an xpath query, let us look at the format of a notification message. A typical notification message resembles the following:

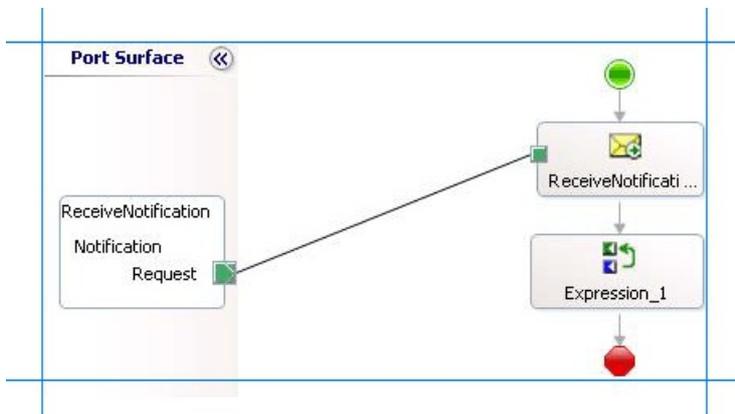
```
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Insert</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

3. As you see, the information about the type of the notification is available within the `<info>` tag, within the parent `<Notification>` tag. So, add the following xpath query within the **Expression** shape:

```
NotificationType = xpath(NotifyReceive, "string(/*[local-name()='Notification']/*[local-name()='Info']/text())");
```

Here, **NotificationType** is the variable you created to store the value extracted by the xpath query. **NotifyReceive** is the message you created in [Step 2: Create Messages for BizTalk Orchestrations](#) to receive notification messages.

4. The following figure shows the in-progress orchestration with the **Expression** shape included.



What did I just do?

In this step, you added an **Expression** shape to extract the kind of notification received from the SQL Server database.

Next Steps

You add a Decide shape to filter for Insert notifications, as described in [Step 3: Add a Filter for Insert Notifications](#).

See Also

Tasks

[Step 1: Add Orchestration Shapes to Receive Notification](#)

[Step 3: Add a Filter for Insert Notifications](#)

Other Resources

[Lesson 2: Receive and Filter Notifications](#)

Step 3: Add a Filter for Insert Notifications



Step 3

Time to complete: 5 minutes

Objective: In this step, you add a Decide shape to the orchestration to filter for notification messages for Insert operation. Subsequent operations in the orchestration are performed only if the notification received is of Insert type.

Prerequisites

You must have completed [Step 2: Extract Notification Type from Notification Message](#).

To filter for notification messages

1. Add a **Decide** shape to the orchestration, after the **Expression** shape. From the Toolbox, drag the **Decide** shape onto the connecting line directly below the **Expression** shape.

The **Decide** shape expands to show a branch for the **If** statement (**Rule_1**) and a branch for the **Else** statement.

2. On the design surface, right-click the **Decide** shape, and then click **Properties Window**.
3. In the **Properties** pane for the **Decide** shape, in the **Name** property, type **CheckNotification**.
4. On the design surface, right-click the **Rule_1** shape (inside of the **Decide** shape), and then click **Properties Window**.
5. In the **Properties** pane for **Rule_1**, in the **Name** property, type **Insert**.
6. Right-click the **Insert** shape, and then click **Edit Boolean Expression**.
7. In the BizTalk Expression Editor, type the following:

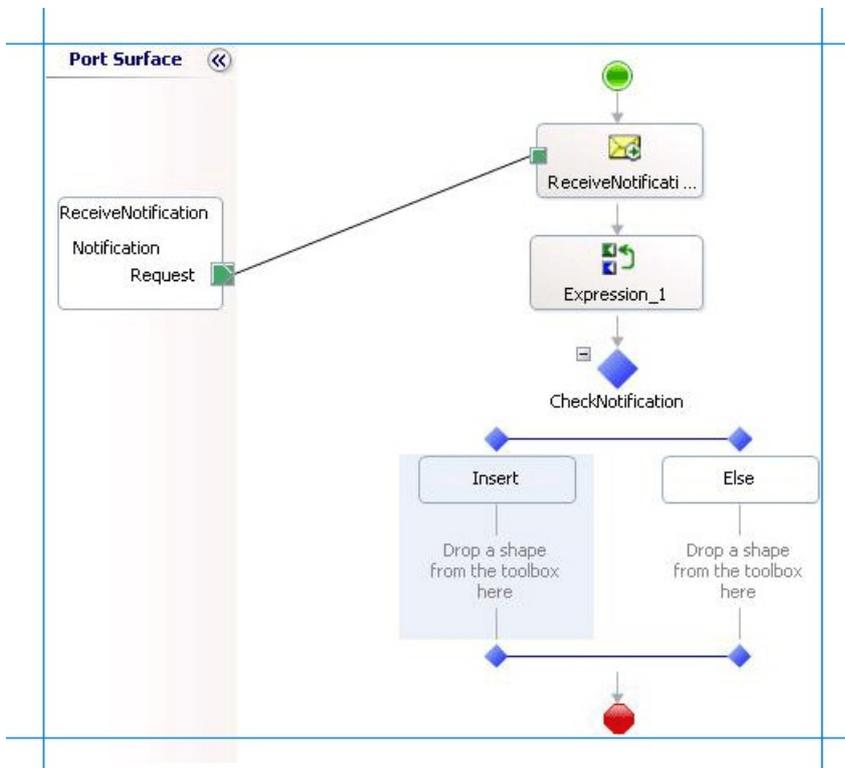
```
NotificationType.Equals("Insert")
```

This condition tells the orchestration to perform subsequent operations only if the value in the **NotificationType** variable is **Insert**.

Note

You added this variable in [Step 2: Extract Notification Type from Notification Message](#) to extract the type of notification from the notification message received from the SQL Server database.

8. The following figure shows the in-progress orchestration with the **Decide** shape included.



What did I just do?

In this step, you added a **Decide** shape to filter the notification messages to perform subsequent operations only if the notification received is for Insert operations.

Next Steps

In the next step, you add orchestration shapes to invoke the UPDATE_EMPLOYEE stored procedure on the Employee table, as described in [Lesson 3: Execute a Stored Procedure to Select New Employees Added](#).

See Also

Tasks

[Step 2: Extract Notification Type from Notification Message](#)

Other Resources

[Lesson 2: Receive and Filter Notifications](#)

Lesson 3: Execute a Stored Procedure to Select New Employees Added

Before understanding the tasks performed in this lesson, you must first understand why these tasks are required. The **Employee** table to which the records are inserted to add a new employee is defined in such a way that a **Status** column is always set to "0" every time a new employee is added. This is done so that you can use this column to query for newly added employees and also get notifications. In SQL Server, you would query this by running the following SQL statement:

```
SELECT Employee_ID, Name, Designation FROM Employee WHERE Status = 0
```

After receiving the list of newly added employees, you must also update the **Status** column to "1" so that the next time new employees are added and you run the same query, you do not get records for old employees as well. To make sure that the above Select statement gives only the newly added employees, you will update the **Employee** table using the following SQL statement:

```
UPDATE Employee SET Status = 1 WHERE Status = 0
```

So, the **Status** column for the old employees is set to "1" while new employees will always be "0."

In this lesson, you will execute a stored procedure, **UPDATE_EMPLOYEE**, which in turn executes the Select and Update statements. After you have finished this lesson, your orchestration will do the following:

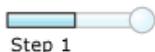
1. Receives notification for any changes to the **Employee** table.
2. Extracts the type of notification from the notification message received.
3. If the notification message is for an Insert operation, the orchestration executes the **UPDATE_EMPLOYEE** stored procedure.
4. The stored procedure reads the newly entered records in the **Employee** table. After reading the new records, the stored procedure also sets the **Status** column for those records to "1."

Now you know why you need to execute the stored procedure. You now need to think about how to execute this as part of the orchestration. The orchestration needs a request message for the **UPDATE_EMPLOYEE** stored procedure. In this tutorial, you will create a custom class library that will create the message on the fly and then provide it to the orchestration. After the orchestration receives the message, it will send the message to the SQL Server using the SQL adapter and receive the response.

In This Section

- [Step 1: Create the Request Message for UPDATE_EMPLOYEE Stored Procedure](#)
- [Step 2: Send the Request Message to SQL Server and Receive Response](#)

Step 1: Create the Request Message for UPDATE_EMPLOYEE Stored Procedure



Time to complete: 10 minutes

Objective: In this step, you add a C# class library project to your solution. This library creates an in-memory request message for the **UPDATE_EMPLOYEE** stored procedure. In later steps, the orchestration sends this message to SQL Server to execute the stored procedure.

Prerequisites

You must have completed the steps in [Lesson 2: Receive and Filter Notifications](#).

To create a request message for UPDATE_EMPLOYEE stored procedure

1. Add a Visual C# class library project to your solution. For the name of the project, type **UpdateEmployeeMessageCreator**.
2. Rename **Class1.cs** to **UpdateEmployeeMessageCreator.cs**.
3. Copy the following code to the .cs file:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.IO;

namespace UpdateEmployeeMessageCreator
{
    public class UpdateEmployeeMessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;

        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\CreateEmployeeMessage";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOptions.TopDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }

            //Create Message From XML
            Message = new XmlDocument();

            Message.PreserveWhitespace = true;

            Message.Load(ResponseDoc);
        }
    }
}
```

```
        return Message;
    }
}
```

This code snippet expects a request message for the **UPDATE_EMPLOYEE** stored procedure to be present at C:\TestLocation\CreateEmployeeMessage. The code uses the request message to create a similar request message at run time.

4. Add a strong-name key file to the project. For instructions on creating a strong-name key file, see [Create a Strong-Name Key File](#).
 - a. In the Solution Explorer, right-click the **UpdateEmployeeMessageCreator** project, and then click **Properties**.
 - b. In the **Property** window, click **Signing**.
 - c. In the **Signing** tab, select the **Sign the assembly** check box.
 - d. From the **Choose a strong-name key file** list, click **<Browse>**.
 - e. Navigate to the folder where you created the strong-name key file, and then click **Open**.
 - f. Click **Save** on the Standard menu bar. Close the **Property** window.
5. Build the project. Right-click the project, and then click **Build**.
6. Add a reference of this project to the BizTalk project in the solution.
 - a. In the Solution Explorer, expand the BizTalk project, right-click **References**, and then click **Add Reference**.
 - b. In the **Add Reference** dialog box, click the **Projects** tab.
 - c. From the list of project names, select **UpdateEmployeeMessageCreator**, click **Add**, and then click **OK**.
7. Building the project creates the assembly DLL under \bin\Debug folder of the project. You must add this DLL to the Global Assembly Cache (GAC).
 - a. Open a Visual Studio command prompt. Click **Start**, click **All Programs**, click **Microsoft Visual Studio 2010**, click **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
 - b. From the command prompt, navigate to the \bin\Debug\ folder of the **UpdateEmployeeMessageCreator** project.
 - c. Run the following command on the command prompt:

```
gacutil /i UpdateEmployeeMessageCreator.dll
```

What did I just do?

In this step, you added an UpdateEmployeeMessageCreator class library project that creates request message at run-time. You added the reference to this project in the BizTalk project and also added the assembly DLL to the GAC.

Next Steps

You send the request message to SQL Server and receive the response, as described in [Step 2: Send the Request Message to SQL Server and Receive Response](#).

See Also

Other Resources

Lesson 3: Execute a Stored Procedure to Select New Employees Added

Step 2: Send the Request Message to SQL Server and Receive Response



Time to complete: 10 minutes

Objective: In this step, you send the request message to execute the **UPDATE_EMPLOYEE** stored procedure and receive the response.

Prerequisites

You must have completed [Step 1: Create the Request Message for UPDATE_EMPLOYEE Stored Procedure](#).

To send the request message and receive a response

1. To the existing orchestration, under the **Insert** block of the **Decide** shape, add a **Message Assignment** shape. From the Toolbox, drag the **Message Assignment** shape to the space indicated.

Note

When you drop the **Message Assignment** shape onto the design surface, Orchestration Designer creates the enclosing **Construct Message** shape for you.

2. On the design surface, right-click the **ConstructMessage_1** shape, and then click **Properties Window**.
3. In the **Properties** pane for the **ConstructMessage_1** shape, specify the following values.

Set this property	To this value
Messages Constructed	UpdateEmployee
Name	ConstructRequestMessage

4. Double-click the **MessageAssignment** shape to open the **BizTalk Expression Editor**.
5. In the **BizTalk Expression Editor**, add the following:

```
UpdateEmployee = UpdateEmployeeMessageCreator.UpdateEmployeeMessageCreator.XMLMessageCreator();
UpdateEmployee(WCF.Action) = "TypedProcedure/dbo/UPDATE_EMPLOYEE";
```

Here, **UpdateEmployee** is the message you created in [Step 2: Create Messages for BizTalk Orchestrations](#) for sending request messages for **UPDATE_EMPLOYEE** stored procedure. In the **MessageAssignment** shape, you invoke the **UpdateEmployeeMessageCreator** class to create a request message. Also, you set the WCF action for the request message.

6. Add the following shapes to the orchestration under the **Message Assignment** shape.

Shape	Shape Type	Properties
SendUpdateMessage	Send	<ul style="list-style-type: none"> • Set Message to <i>UpdateEmployee</i> • Set Name to <i>SendUpdateMessage</i>

ReceiveUpdateResponse	Receive	<ul style="list-style-type: none"> • Set Activate to <i>False</i> • Set Message to <i>UpdateEmployeeResponse</i> • Set Name to <i>ReceiveUpdateResponse</i>
-----------------------	---------	---

7. Add a request-response send port to the orchestration. You will use this port to send request messages to the SQL Server and receive response. Set the following properties for the port.

Set this property	To this value
Communication Direction	Send-Receive
Communication Pattern	Request-Response
Identifier	SQLOutboundPort

Also, change the operation name from Operation_1 to **UpdateEmp**.

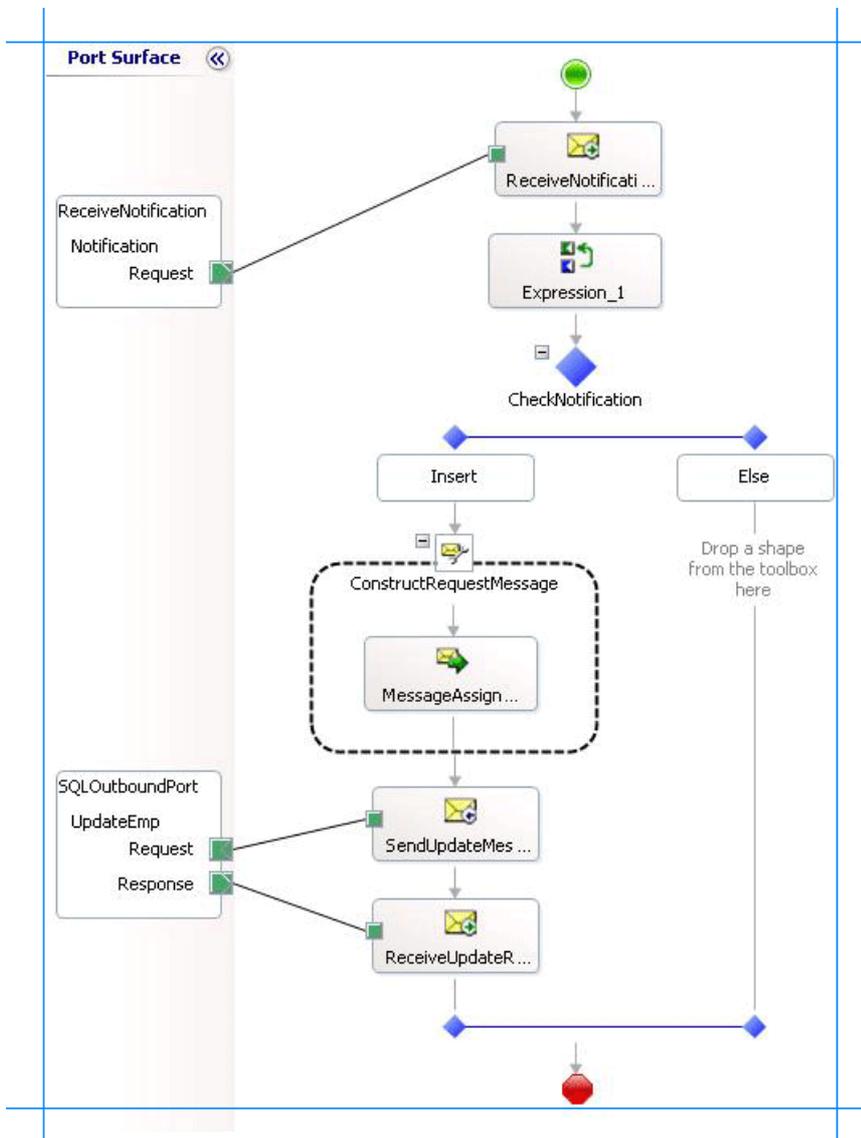
8. Connect the port to action shapes. In Orchestration Designer, on the design surface, drag the green arrow-shaped handle for the port to the corresponding green handle of the action shape.

<p>Note</p> <p>In this step, you use the drag-and-drop method to connect ports to action shapes. You could instead use the operation property of an action shape to connect the action shape to a port.</p>
--

Connect the ports and action shapes as follows:

- Connect the **SendUpdateMessage** action shape to the **Request** handle of the **SQLOutboundPort**.
- Connect the **ReceiveUpdateResponse** action shape to the **Response** handle of the **SQLOutboundPort**.

9. The following figure shows the in-progress orchestration.



What did I just do?

In this step, you updated the orchestration by adding a **MessageAssignment** shape, **Send** and **Receive** shapes, and a port. You connected the shapes and ports to send request message to execute the `UDPATE_EMPLOYEE` request message and receive the response.

Next Steps

In the next step, you add orchestration shapes to invoke the `Insert` operation on the **Purchase_Order** table, as described in [Lesson 4: Perform an Insert Operation on the Purchase Order Table](#).

See Also

Tasks

[Step 1: Create the Request Message for UPDATE_EMPLOYEE Stored Procedure](#)

Other Resources

[Lesson 3: Execute a Stored Procedure to Select New Employees Added](#)

Lesson 4: Perform an Insert Operation on the Purchase Order Table

In [Lesson 3: Execute a Stored Procedure to Select New Employees Added](#), you executed the **UPDATE_EMPLOYEE** stored procedure and received a response message that contains the details of the newly inserted employee record. In this lesson, you will build on the previous lesson and update the orchestration to perform the following steps:

1. Within the orchestration, you build the message to perform an Insert operation on the **Purchase_Order** table. This step is similar to [Step 1: Create the Request Message for UPDATE_EMPLOYEE Stored Procedure](#).
2. After you build the request message, you map the response message of the **UPDATE_EMPLOYEE** stored procedure to the request message for the Insert operation on the **Purchase_Order** table. By mapping the messages, you pass the values received from the response messages to the request message for Insert operation.
3. You send the message to insert a record in the **Purchase_Order** table and receive a response.
4. You send the response to a send port.

In This Section

- [Step 1: Create the Request Message for Insert Operation on Purchase_Order Table](#)
- [Step 2: Map the UPDATE_EMPLOYEE Response Message to Insert Operation Request Message](#)
- [Step 3: Send the Request Message to Insert Records and Receive a Response](#)
- [Step 4: Build the Project](#)

Step 1: Create the Request Message for Insert Operation on Purchase_Order Table



Time to complete: 10 minutes

Objective: In this step, you add a C# class library project to your solution. This library creates an in-memory request message for the Insert operation on the **Purchase_Order** table. In later steps, the orchestration sends this message to SQL Server to insert records in the table.

Prerequisites

You must have completed the steps in [Lesson 3: Execute a Stored Procedure to Select New Employees Added](#).

To create a request message for Insert operation

1. Add a Visual C# class library project to your solution. For the name of the project, type **UpdatePOMessageCreator**.
2. Rename **Class1.cs** to **UpdatePOMessageCreator.cs**.
3. Copy the following code to the .cs file:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using System.IO;

namespace UpdatePOMessageCreator
{
    public class UpdatePOMessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;

        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\\\CreatePOMessage";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOptions.TopDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }

            //Create Message From XML
            Message = new XmlDocument();

            Message.PreserveWhitespace = true;

            Message.Load(ResponseDoc);
        }
    }
}
```

```
        return Message;
    }
}
```

This code snippet expects a request message for the Insert operation on the **Purchase_Order** table to be present at C:\TestLocation\CreatePOMessage. The code uses the request message to create a similar request message at run time.

4. Add a strong-name key file to the project. For instructions on creating a strong-name key file, see [Create a Strong-Name Key File](#).
 - a. In the Solution Explorer, right-click the **UpdatePOMessageCreator** project and click **Properties**.
 - b. In the **Property** window, click **Signing**.
 - c. In the **Signing** tab, select the **Sign the assembly** check box.
 - d. From the **Choose a strong-name key file** list, click **<Browse>**.
 - e. Navigate to the folder where you created the strong-name key file, and then click **Open**.
 - f. Click **Save** on the **Standard** menu bar. Close the **Property** window.
5. Build the project. Right-click the project and click **Build**.
6. Add a reference of this project to the BizTalk project in the solution.
 - a. In the Solution Explorer, expand the BizTalk project, right-click **References**, and then click **Add Reference**.
 - b. In the **Add Reference** dialog box, click the **Projects** tab.
 - c. From the list of project names, select **UpdatePOMessageCreator**, click **Add**, and then click **OK**.
7. Building the project creates the assembly DLL under \bin\Debug folder of the project. You must add this DLL to the Global Assembly Cache (GAC).
 - a. Open a Visual Studio command prompt. Click **Start**, click **All Programs**, click **Microsoft Visual Studio 2010**, click **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
 - b. From the command prompt, navigate to the \bin\Debug\ folder of the **UpdatePOMessageCreator** project.
 - c. Run the following command on the command prompt:

```
gacutil /i UpdatePOMessageCreator.dll
```

What did I just do?

In this step, you added an UpdatePOMessageCreator class library project that creates request message at run-time. You added the reference to this project in the BizTalk project and also added the assembly DLL to the GAC.

Next Steps

You map the response message for the UPDATE_EMPLOYEE stored procedure to the request message for the Insert operation on **Purchaser_Order** table.

See Also

Tasks

[Step 2: Map the UPDATE_EMPLOYEE Response Message to Insert Operation Request Message](#)

Other Resources

Lesson 4: Perform an Insert Operation on the Purchase Order Table

Step 2: Map the UPDATE_EMPLOYEE Response Message to Insert Operation Request Message



Time to complete: 10 minutes

Objective: In this step, you create the request message to perform an Insert operation on the **Purchase_Order** table and then map the response message for the **UPDATE_EMPLOYEE** stored procedure to the request message for the Insert operation. By doing so, you pass on the values in the response message to be inserted in the **Purchase_Order** table.

Prerequisites

You must have completed [Step 1: Create the Request Message for Insert Operation on Purchase_Order Table](#).

To map the messages

1. To the existing orchestration, in the **Insert** block of the **Decide** shape, under the **ReceiveUpdateResponse** shape, add a **Message Assignment** shape. From the Toolbox, drag the **Message Assignment** shape to the space indicated.

Note

When you drop the **Message Assignment** shape onto the design surface, Orchestration Designer creates the enclosing **Construct Message** shape for you.

2. On the design surface, right-click the **ConstructMessage_1** shape, and then click **Properties Window**.
3. In the **Properties** pane for the **ConstructMessage_1** shape, specify the following values.

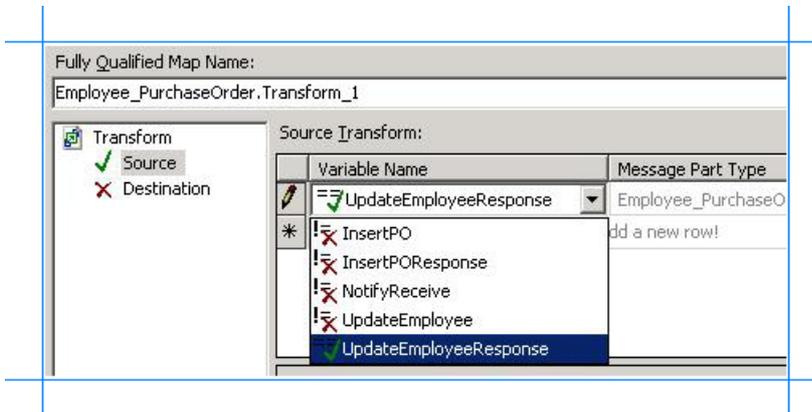
Set this property	To this value
Messages Constructed	InsertPO
Name	ConstructInsertMessage

4. Double-click the **MessageAssignment** shape to open the **BizTalk Expression Editor**.
5. In the **BizTalk Expression Editor**, add the following:

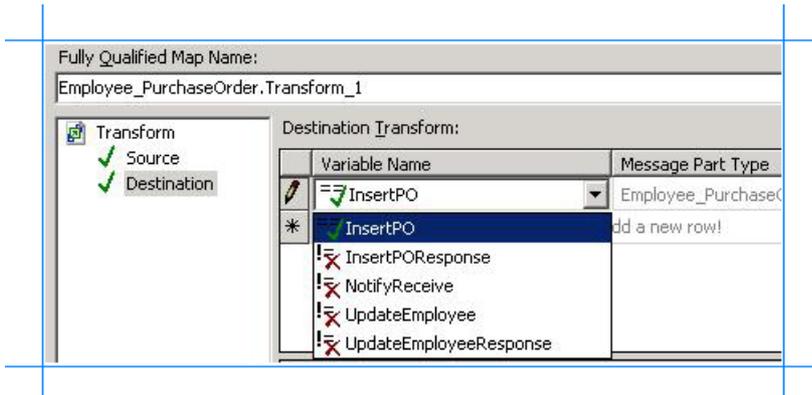
```
InsertPO = UpdatePOMessageCreator.UpdatePOMessageCreator.XMLMessageCreator();
InsertPO(WCF.Action) = "TableOp/Insert/dbo/Purchase_Order";
```

Here, **InsertPO** is the message you created in [Step 2: Create Messages for BizTalk Orchestrations](#) for sending request messages for Insert operation on the **Purchase_Order** table. In the **MessageAssignment** shape, you invoke the **UpdatePOMessageCreator** class to create a request message. Also, you set the WCF action for the request message.

6. Within the **Construct Message** shape and after the **Message Assignment** shape, add a **Transform** shape.
7. In the **Transform Configuration** dialog box, from the left pane, under the **Transform** label, click **Source**.
8. From the **Source Transform** box on the right, click the space under the **Variable Name**, and then select **UpdateEmployeeResponse**.

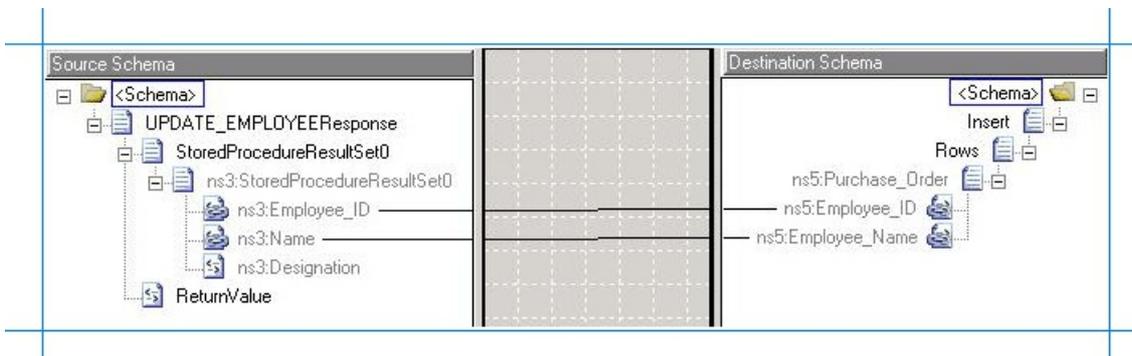


9. In the **Transform Configuration** dialog box, from the left pane, under the **Transform** label, click **Destination**.
10. From the **Destination Transform** box on the right, click the space under the **Variable Name**, and then select **InsertPO**.



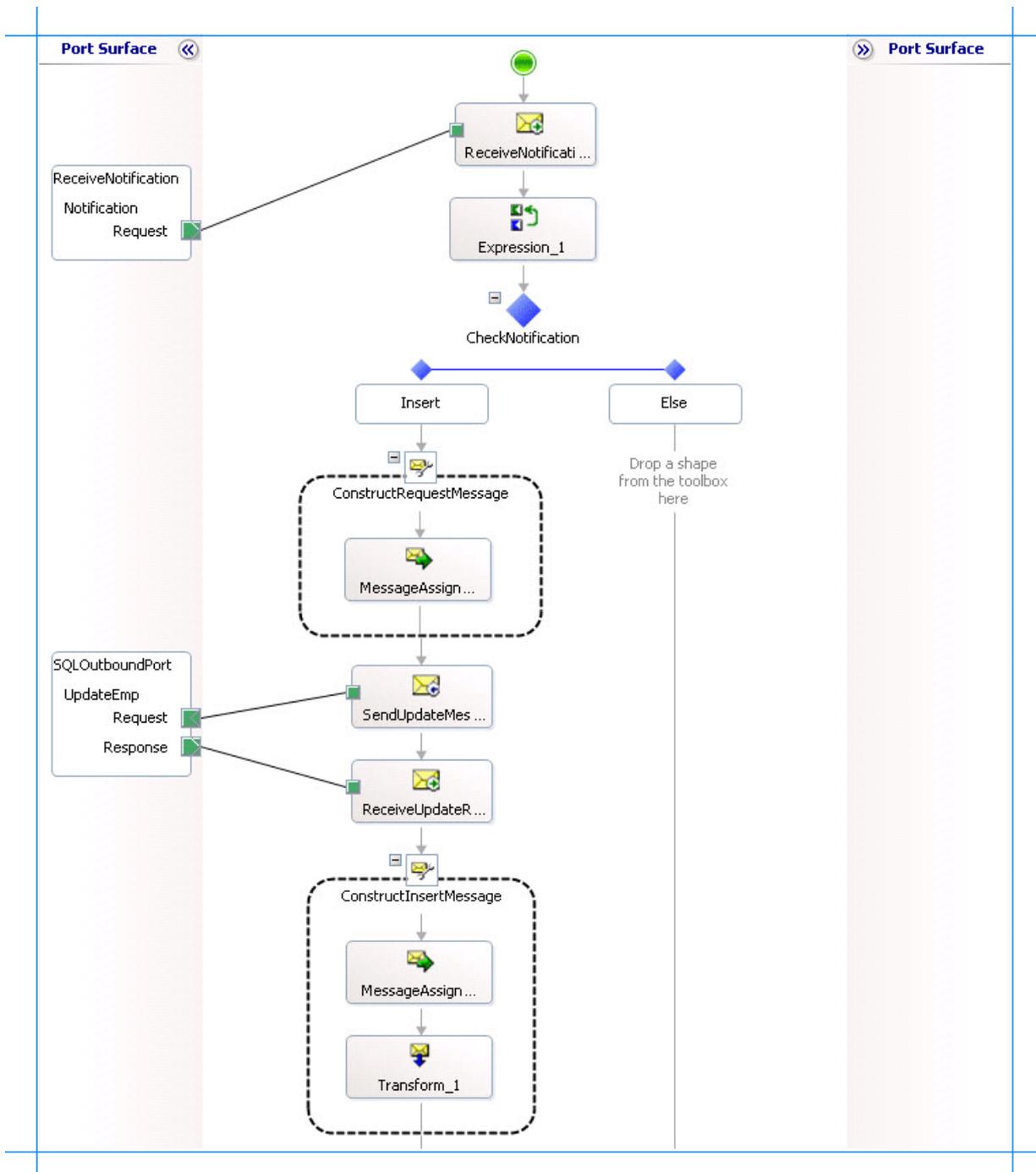
11. Click **OK**. The map file opens.
12. Expand the nodes in the source and destination schemas.
13. Map the Employee_ID and name fields in both the schemas.
 - Map the **Employee_ID** node in the source schema (UPDATE_EMPLOYEEEResponse) to the **Employee_ID** node in the destination schema (Insert).
 - Map the **Name** node in the source schema to the **Employee_Name** in the destination schema.

The following figure shows the mapped schemas.



Save and close the map.

14. The following figure shows the in-progress orchestration.



What did I just do?

In this step, you created a message to insert records into the **Purchase_Order** table and then mapped the response message from the **UPDATE_EMPLOYEE** stored procedure to the request message for the Insert operation.

Next Steps

You send the request message to perform an Insert operation on the **Purchase_Order** table and receive a response, as described in [Step 3: Send the Request Message to Insert Records and Receive a Response](#).

See Also

Tasks

[Step 1: Create the Request Message for Insert Operation on Purchase_Order Table](#)

Other Resources

[Lesson 4: Perform an Insert Operation on the Purchase Order Table](#)

Step 3: Send the Request Message to Insert Records and Receive a Response



Time to complete: 10 minutes

Objective: In this step, you send the request message to insert records into the **Purchase_Order** table and receive a response.

Prerequisites

You must have completed [Step 2: Map the UPDATE_EMPLOYEE Response Message to Insert Operation Request Message](#).

To send the request message and receive a response

1. Add the following shapes to the orchestration under the **Construct Message** shape.

Shape	Shape Type	Properties
SendInsertMessage	Send	<ul style="list-style-type: none"> • Set Message to <i>InsertPO</i> • Set Name to <i>SendInsertMessage</i>
ReceiveInsertResponse	Receive	<ul style="list-style-type: none"> • Set Activate to <i>False</i> • Set Message to <i>InsertPOResponse</i> • Set Name to <i>ReceiveInsertResponse</i>
SaveInsertResponse	Send	<ul style="list-style-type: none"> • Set Message to <i>InsertPOResponse</i> • Set Name to <i>SaveInsertResponse</i>

2. Modify the **SQLOutboundPort** you created in [Step 2: Send the Request Message to SQL Server and Receive Response](#).
 - a. Right-click the port in the Orchestration Designer, and then click **New Operation**. The port shape changes to add a new operation, **Operation_1**.
 - b. Click **Operation_1** and in the properties window, change the value of Identifier to **InsertPO**.
3. Add a one-way send port to the orchestration. You will use this port to send the response message for the Insert operation. Set the following properties for the port.

Set this property	To this value
Communication Direction	Send
Communication Pattern	One-Way
Identifier	SaveResponsePort

Also, change the operation name from Operation_1 to **InsertPO**.

4. Connect the port to action shapes. In Orchestration Designer, on the design surface, drag the green arrow-shaped handle for the port to the corresponding green handle of the action shape.

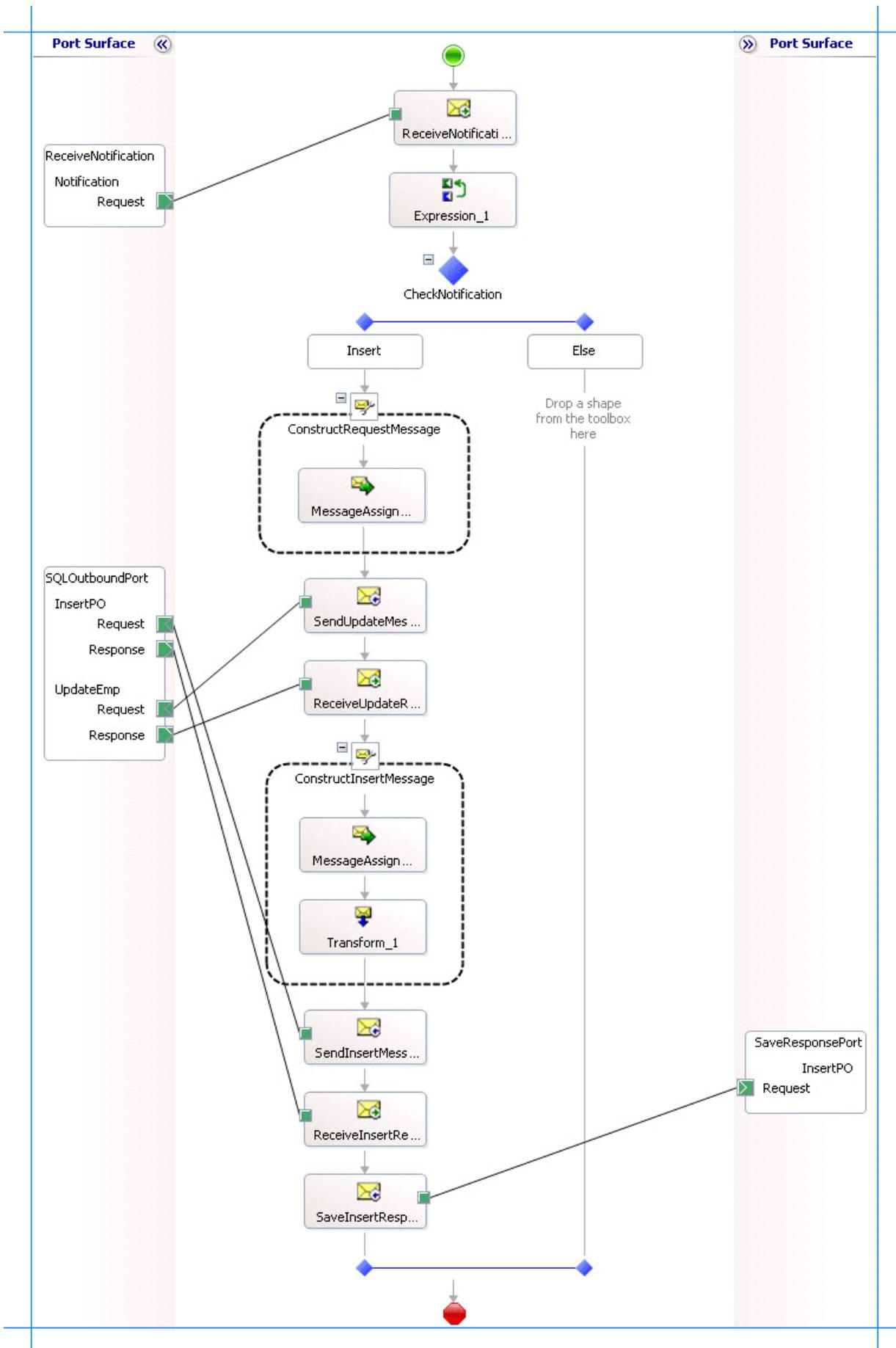
Note

In this step, you use the drag-and-drop method to connect ports to action shapes. You could instead use the operation property of an action shape to connect the action shape to a port.

Connect the ports and action shapes as follows:

- Connect the **SendInsertMessage** action shape to the **Request** handle of the **InsertPO** operation of the **SQLOutboundPort**.
- Connect the **ReceiveInsertResponse** action shape to the **Response** handle of the **InsertPO** operation of the **SQLOutboundPort**.
- Connect the **SaveInsertResponse** action shape to the **Request** handle of the **SaveResponsePort**.

5. The following figure shows the in-progress orchestration.



What did I just do?

You sent the request to insert records into the **Purchase_Order** table and receive a response.

Next Steps

You build the project, as described in [Step 4: Build the Project](#).

See Also

Tasks

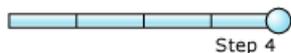
[Step 2: Map the UPDATE_EMPLOYEE Response Message to Insert Operation Request Message](#)

[Step 4: Build the Project](#)

Other Resources

[Lesson 4: Perform an Insert Operation on the Purchase Order Table](#)

Step 4: Build the Project



Time to complete: 5 minutes

Objective: In this step, you compile the BizTalk orchestration project.

Prerequisites

You must have completed [Step 3: Send the Request Message to Insert Records and Receive a Response](#).

To build the BizTalk orchestration project

1. In the Solution Explorer, right-click the BizTalk project name, and then click **Properties**.
2. In the property pages dialog box, in the tree pane, expand **Common Properties**, click **Assembly**, and then in the properties list, click the **Assembly Key File** ellipsis [...].
3. Specify a path to the assembly key file you created as described in [Create a Strong-Name Key File](#), and then click **Open**.
4. In the property pages dialog box, in the tree pane, expand **Configuration Properties**, click **Deployment**, and then do the following:
 - a. For the **Application Name** property, type **SampleApplication**.
 - b. For the **Redeploy** property, select **True**.

Click **OK**.

5. On the **File** menu, click **Save All**.
6. In Solution Explorer, right-click the solution name, and then click **Build Solution**.

The Output pane at the bottom of the screen should read: **Build: 3 succeeded or up-to-date, 0 failed, 0 skipped**.

What did I just do?

In this step, you compiled the solution containing the BizTalk project and two class library projects.

Next Steps

You deploy the solution, as described in [Lesson 5: Deploy the Solution](#).

See Also

Tasks

[Step 3: Send the Request Message to Insert Records and Receive a Response](#)

Other Resources

[Lesson 4: Perform an Insert Operation on the Purchase Order Table](#)

Lesson 5: Deploy the Solution

In this lesson, you deploy the BizTalk solution using the BizTalk Server Administration console. In the BizTalk orchestration designer, you created logical ports. You will now create the physical ports and then configure the BizTalk application by mapping the logical ports to the physical ports.

After you have configured the application, you will start and test the application.

In This Section

- [Step 1: Deploy the Orchestration](#)
- [Step 2: Configure the Ports](#)
- [Step 3: Configure and Start the Application](#)
- [Step 4: Test the Application](#)

Step 1: Deploy the Orchestration



Step 1

Time to complete: 5 minutes

Objective: In this step, deploy the orchestration solution.

Prerequisites

You must have completed the steps in [Lesson 4: Perform an Insert Operation on the Purchase Order Table](#).

To deploy the solution

1. In Solution Explorer, right-click the solution name, and then click **Properties**.
2. In the properties pages dialog box, in the tree control, expand **Configuration Properties**, and then click **Configuration**.
3. On the **Configuration** page, in the **Deploy** column, select the check box against the BizTalk project, and then click **OK**.
4. In Solution Explorer, right-click the solution name, and then click **Deploy Solution**.

The Output pane at the bottom of the screen should read: **Deploy: 1 succeeded, 0 failed, 0 skipped**.

What did I just do?

In this step, you deployed the BizTalk orchestration to BizTalk Server Administration console.

Next Steps

You create the physical ports in BizTalk Server Administration console, as described in [Step 2: Configure the Ports](#).

See Also

Tasks

[Step 2: Configure the Ports](#)

Other Resources

[Lesson 5: Deploy the Solution](#)

Step 2: Configure the Ports



Time to complete: 15 minutes

Objective: In this step, you create the physical ports in BizTalk Server Administration console. You create a physical port for each logical port you created in the orchestration. You will create the following ports:

- A one-way WCF-Custom receive port to receive notification messages for changes to **Employee** table in a SQL Server database.
- A request-response WCF-Custom send port to send request messages and receive response for invoking the **UPDATE_EMPLOYEE** stored procedure and for performing the Insert operation on the **Purchase_Order** table. In the orchestration, you used the same send port to perform both the operations. Similarly, in the BizTalk Server Administration console, you will use a single send port for both operations.
- A one-way send port to send the response for the Insert operation. In this tutorial, because you need to inform the Purchases department through an e-mail, you create this send port as an SMTP port.

Prerequisites

You must have completed [Step 1: Deploy the Orchestration](#).

To create a physical one-way receive port

1. Start the BizTalk Server Administration console.
2. In the console tree on the left hand side, expand **BizTalk Server Administration**, right-click **BizTalk Group**, and then click **Refresh**.
3. Expand **BizTalk Group**, expand **Applications**, and expand **SampleApplication**. For this tutorial, you create all the ports and application within the SampleApplication application.
4. Follow the instructions under the "Deploying Adapters for Receiving Messages from SQL Server" section of [Configuring a Port Using the WCF-Custom Adapter](#). Name the port as **NotifyReceivePort**.
5. Make sure you set the following binding properties to configure the adapter to receive notifications for changes to the **Employee** table.

Binding property	Value
InboundOperationType	Set this to Notification .
NotificationStatement	Set this to: <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0</pre> </div> <p> Note You must specifically specify the column names in the statement as shown in this Select statement. Also, you must always specify the table name along with the schema name, for example, <code>dbo.Employee</code>.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

To create a request-response send port for two operations

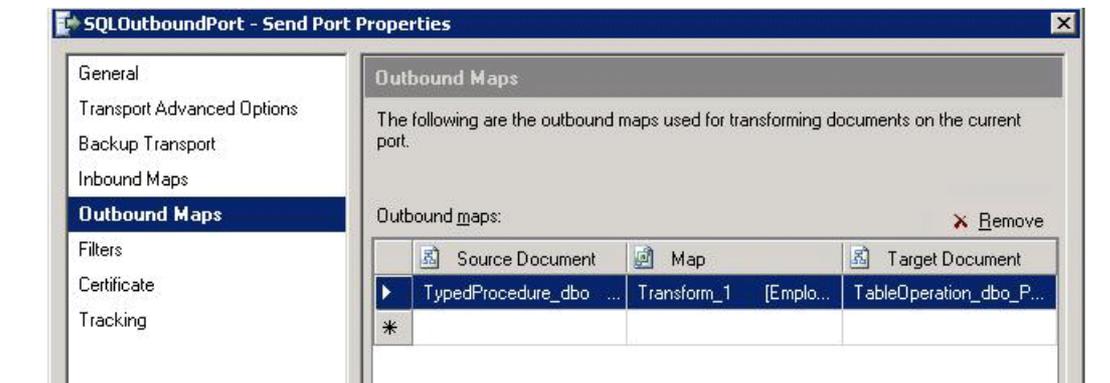
1. Follow the instructions under the "Deploying Adapters for Sending Messages to SQL Server" section of [Configuring a Port Using the WCF-Custom Adapter](#). Name the port as **SQLOutboundPort**.
2. Because you are performing two operations using the same send port, you must use dynamic action mapping to specify the action for the operation. While configuring the port, in the **Action** box, specify the action mapping in the following manner:

```
<BtsActionMapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Operation Name="UpdateEmp" Action="TypedProcedure/dbo/UPDATE_EMPLOYEE" />
  <Operation Name="InsertPO" Action="TableOp/Insert/dbo/Purchase_Order" />
</BtsActionMapping>
```

Note that in the orchestration, you created two operations for the request-response send port: **UpdateEmp** and **InsertPO**. So, in the physical port configuration you provide the same operation names in the dynamic action mapping. In the above excerpt, the action for **UpdateEmp** operation is `TypedProcedure/dbo/UPDATE_EMPLOYEE`. Similarly, the action for **InsertPO** operation is `TableOp/Insert/dbo/Purchase_Order`.

3. You must also configure the send port to use the Mapper you created in the orchestration to map the response message of **UPDATE_EMPLOYEE** stored procedure to the request message for the Insert operation on **Purchase_Order** table. To do so:
 - a. Right-click the **SQLOutboundPort** in BizTalk Server Administration console, and then click **Properties**.
 - b. From the **SQLOutboundPort – Send Port Properties** dialog box, from the left pane, click **Outbound Maps**.
 - c. From the right-pane, in the **Outbound Maps** box, click the cell under the **Map** column, and from the drop-down list, select **Transform_1**. This is the name of the map you created in the BizTalk orchestration in Visual Studio.

Click **OK**.



To create an SMTP send port

1. Follow the instructions under the "How to Configure an SMTP Send Port with the BizTalk Server Administration Console" section at <http://go.microsoft.com/fwlink/?LinkId=141549>. Name the port as **EmailResponse**.
2. As part of the port configuration, specify the e-mail address for the Purchases department for the **To** property.

What did I just do?

In this step you created a WCF-Custom receive port for receiving notifications from SQL Server, WCF-Custom send port for performing operations on SQL Server, and an SMTP port for sending the response from SQL Server as an e-mail to the Purchases department.

Next Steps

You configure and start the BizTalk application, as described in [Step 3: Configure and Start the Application](#).

See Also

Tasks

[Step 1: Deploy the Orchestration](#)

[Step 3: Configure and Start the Application](#)

Other Resources

[Lesson 5: Deploy the Solution](#)

Step 3: Configure and Start the Application



Time to complete: 10 minutes

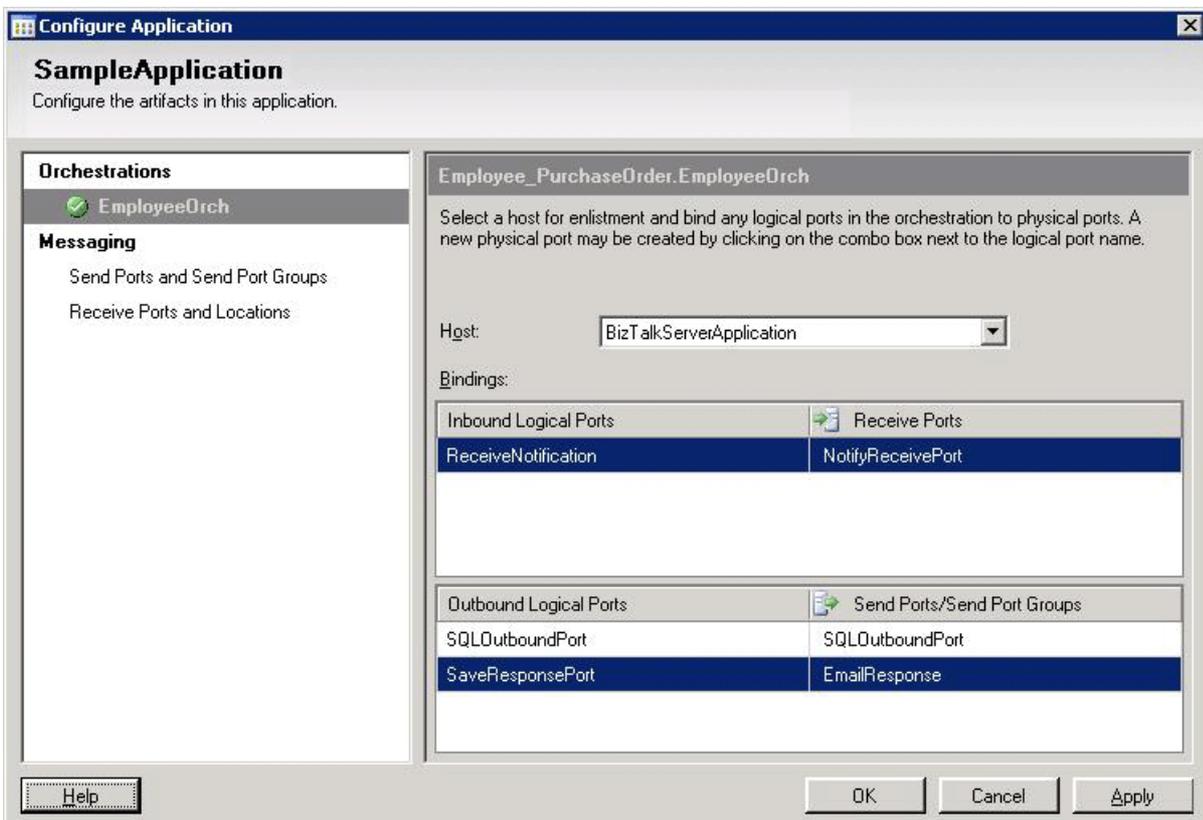
Objective: In this step, you configure and start the SampleApplication application. When you configure the SampleApplication application, you associate the logical artifacts you created in Visual Studio with their physical counterparts.

Prerequisites

You must have completed [Step 2: Configure the Ports](#).

To configure and start the application

1. Start the BizTalk Server Administration console.
2. In the console tree on the left hand side, expand **BizTalk Server Administration**, right-click **BizTalk Group**, and then click **Refresh**.
3. Expand **BizTalk Group**, expand **Applications**, right-click **SampleApplication**, and then click **Configure**.
4. In the **Configure Application** dialog box, on the **EmployeeOrch** tab, do the following:
 - a. For **Host** drop-down list, select **BizTalkServerApplication**.
 - b. Double-click the cell across **ReceiveNotification** and select **NotifyReceivePort** from the drop-down list.
 - c. Double-click the cell across **SQLOutboundPort** and select **SQLOutboundPort** from the drop-down list.
 - d. Double-click the cell across **SaveResponsePort** and select **EmailResponse** from the drop-down list.
5. The following figure shows a configured application.



6. In the **Configure Application** dialog box, click **OK**.
7. In the console tree, right-click **SampleApplication**, and then click **Start**.

8. In the console tree, click **Applications**.

9. In the Applications details pane, check that the **Status** of **SampleApplication** is **Started**.

What did I just do?

You configured and started the SampleApplication application

Next Steps

You test the application by inserting new employees in the **Employee** table, as described in [Step 4: Test the Application](#).

See Also

Tasks

[Step 2: Configure the Ports](#)

[Step 4: Test the Application](#)

Other Resources

[Lesson 5: Deploy the Solution](#)

Step 4: Test the Application



Time to complete: 10 minutes

Objective: In this step, you test the application by inserting a record in the **Employee** table of the **ADAPTER_SAMPLES** database. If the application is working properly, the orchestration receives a notification for changes to the **Employee** table. The orchestration then extracts the type of notification received. If the notification is for an Insert operation, the orchestration executes the **UPDATE_EMPLOYEE** stored procedure and receives a response. The orchestration extracts the values of **Employee_ID** and **Name** from the response and inserts them into the **Purchase_Order** table.

Prerequisites

Before starting with this step, you must ensure the following:

- A request message to invoke the **UPDATE_EMPLOYEE** stored procedure is available at C:\TestLocation\CreateEmployeeMessage. The request message looks like the following:

```
<UPDATE_EMPLOYEE xmlns="http://schemas.microsoft.com/Sql/2008/05/TypedProcedures/dbo" />
```

- A request message to invoke the Insert operation on the **Purchase_Order** table is available at C:\TestLocation\CreatePOMessage. The request message looks like the following:

```
<Insert xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Purchase_Order">
  <Rows>
    <Purchase_Order xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/Tables/dbo">
      <Employee_ID>10</Employee_ID>
      <Employee_Name>Employee_Name</Employee_Name>
    </Purchase_Order>
  </Rows>
</Insert>
```

Note

The values for the **Employee_ID** and **Employee_Name** fields are placeholders. The actual values are inserted by the orchestration at run-time.

- You must have completed [Step 3: Configure and Start the Application](#).

To test the application

1. Insert a record in the **Employee** table. You can do so by running the following statement from SQL Server Management Studio.

```
INSERT INTO [ADAPTER_SAMPLES].[dbo].[Employee] ([Name] ,[Designation] ,[Salary])
VALUES('John Smith' , 'Manager' ,500000)
```

2. Check the **Employee** table in the database. You will notice that the new record is added by the **Status** column is "0."
3. Keep refreshing the **Employee** table records. You will notice that the **Status** column for the new record is now set to "1."
4. Check the **Purchase_Order** table. You will notice that a record with the same employee name and designation, as you provided in the Insert statement, is added to the table.
5. If you provided your e-mail alias in the SMTP port configuration, you will also receive an e-mail with the response message for the Insert operation.

What did I just do?

Tested the SampleApplication application by inserting a record in the **Employee** table.

Next Steps

If the test worked, congratulations! You have completed the SQL adapter tutorial.

If the test did not work, carefully check your work to make sure that you added all of the necessary objects and set their properties correctly.

See Also

Tasks

[Step 3: Configure and Start the Application](#)

Other Resources

[Lesson 5: Deploy the Solution](#)

Community Resources

There are two principal sources of information from Microsoft about the BizTalk Adapter Pack:

- The documentation and samples installed with the BizTalk Adapter Pack.
- The Microsoft BizTalk Adapter Pack forums.

BizTalk Adapter Pack Documentation

You can install the BizTalk Adapter Pack documentation when you install the BizTalk Adapter Pack. (For information about how to install the BizTalk Adapter Pack, see the installation guide. The installation guide is typically available at <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.) The core documentation covers the concepts and procedures required to use the BizTalk Adapter Pack.

After you install the BizTalk Adapter Pack Setup program, you can access the documentation through the **Microsoft BizTalk Adapter Pack** program group on the Start menu.

BizTalk Adapter Pack Samples

This BizTalk Adapter Pack release will provide samples demonstrating how to use the adapters with:

- BizTalk Server.
- The Windows Communication Foundation (WCF) channel model.
- The WCF service model.

In addition, migration samples are provided.

The samples are available at <http://go.microsoft.com/fwlink/?LinkId=196854>.

BizTalk Adapter Pack Forums

You can post your queries about the BizTalk Adapter Pack on the forums.

Resource	Location
BizTalk Adapter Pack Forums	http://go.microsoft.com/fwlink/?LinkId=87695

See Also

Other Resources

[Getting Started](#)

Frequently Asked Questions

This section contains frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for SQL Server and the BizTalk Adapter Pack in general that you might come across while learning or using the product. The FAQs are divided into four categories: General, BizTalk Server, WCF Service Model, and WCF Channel Model.

However, if you have specific questions related to Microsoft BizTalk Adapter for SQL Server or BizTalk Adapter Pack that are not listed in this section, you can post your query on the BizTalk Adapter Pack forum at <http://go.microsoft.com/fwlink/?LinkId=87695>.

In This Section

- [General](#)
- [BizTalk Server](#)
- [WCF Service Model](#)
- [WCF Channel Model](#)

See Also

Other Resources

[Getting Started](#)

General

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for SQL Server and the BizTalk Adapter Pack in general.

What are the supported BizTalk versions for the BizTalk Adapter Pack?

The BizTalk Adapter Pack is supported with Microsoft BizTalk Server 2010.

In which user context should the setup wizard be run?

You must always run the BizTalk Adapter Pack Setup Wizard under a user context with administrative privileges on the computer.

What is WCF? Where can I read more about it?

WCF stands for Windows Communication Foundation. WCF is a programming framework developed by Microsoft for building service-oriented applications. WCF is a part of .NET framework, and enables developers to build secure, reliable, and transacted solutions that integrate across platforms and interoperate with existing investments.

For more information about WCF, see <http://go.microsoft.com/fwlink/?LinkId=19696>.

What is WCF LOB Adapter SDK? Where can I read more about it?

The WCF LOB Adapter SDK is a collection of tools and components that provide a consistent framework for developing reusable, metadata-rich adapters for line-of-business systems. Adapters written using the WCF LOB Adapter SDK are surfaced as custom WCF bindings and can be consumed by a WCF-capable client.

For more information about WCF LOB Adapter SDK, see the WCF LOB Adapter SDK documentation, which is installed along with the WCF LOB Adapter SDK at the following location on your computer: <installation drive>:\Program Files\WCF LOB Adapter SDK\Documents.

How do I get started using the adapter?

To get started with using the SQL adapter, you must figure out a role that suits you as a user of the adapter. The potential roles that are users of the SQL adapter are the following: Developers, IT professionals, and Independent Software Vendors (ISVs). For information about the prerequisites for each of these roles, see [Prerequisite Skills and Knowledge](#). For a list of tasks that are applicable for various users of the SQL adapter, see [Common Administrative Tasks](#) and [Common Developer Tasks](#).

How can I use the SQL adapter to communicate with the SQL Server database?

You can use the SQL adapter to communicate with the SQL Server database either by developing BizTalk applications, using the WCF service model or using the WCF channel model. For more information, see [Overview of BizTalk Adapter for SQL Server](#).

What interfaces are supported by the SQL adapter for retrieving metadata?

The SQL adapter supports two interfaces for retrieving metadata:

- MetadataExchange provided by WCF. WCF provides a metadata-exchange endpoint for all WCF bindings, which enables clients to get metadata from the SQL Server database.
- IMetadataRetrievalContract provided by the WCF LOB Adapter SDK, which supports the metadata browsing and searching capabilities of the adapter.

How does the SQL adapter support high availability of data?

While specifying the [connection URI](#) to connect to a SQL Server database, the SQL adapter allows you to specify the name of a failover SQL Server database to connect to if the primary SQL Server database is not available. The failover SQL Server database is specified using an optional parameter, FailoverPartner, in the connection URI.

Can I migrate BizTalk projects created using the previous version of the SQL adapter to use the WCF-based SQL adapter? How?

Yes. To know the steps for migrating BizTalk projects created using the previous version of the SQL adapter to use the WCF-based SQL adapter, see [SQL Adapter Tutorials](#).

Does the SQL adapter provide a secure way of communicating with the SQL Server database? Are there any best practices to

ensure security of data?

The SQL adapter supports Enterprise Single Sign-On (SSO) and Integrated Security for authentication on the connections that it establishes with the SQL Server database. With SSO, the credentials are encrypted and stored in the registry. The system uses these credentials to determine access instead of requiring the user to enter them where they might be seen by unauthorized actors. Integrated Security uses the credentials of the logged on user to access the SQL server. This also eliminates the need for users to enter credentials. The database administrator must configure SQL to accept users' credentials for Integrated Security to work correctly.

The SQL adapter also does not allow you to enter the user credentials in the connection URI for the SQL Server database while working with the Add Adapter Service Reference Visual Studio Plug-in and Consume Adapter Service BizTalk Project Add-in to prevent credentials from appearing in clear text. Moreover, the password is not written to the configuration file (generated by the Add Adapter Service Reference Visual Studio Plug-in) and the binding file (generated by the Consume Adapter Service BizTalk Project Add-in).

For more information about:

- Data security in the SQL adapter, see [Security and Protection](#).
- Best practices to ensure data security in the SQL adapter, see [Best Practices](#).

Is there a GUI provided by the SQL adapter to view and perform operations on the artifacts in my underlying SQL Server database?

The Consume Adapter Service BizTalk Project Add-in and the Add Adapter Service Reference Visual Studio Plug-in provide a dialog box where you can view and perform operations on the artifacts in the underlying SQL Server database. For more information about the GUI provided by the SQL adapter, see [Browsing, Searching, and Retrieving Metadata for SQL Server Operations](#).

What are binding properties in the SQL adapter? Where can I find information about all the binding properties in SQL adapter?

Adapter clients can use binding properties in the SQL adapter to configure and control the adapter's behavior. For information about all the binding properties surfaced in the SQL adapter, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

What is MSDTC? Do I need to bother about it before using SQL adapter?

MSDTC stands for Microsoft Distributed Transaction Coordinator. MSDTC coordinates various transactions among multiple resource managers such as databases, file systems, and message queues. To use the SQL adapter with BizTalk Server, you must enable MSDTC. For detailed information about configuring MSDTC, see [Configure MSDTC on SQL Server and Adapter Client](#).

Where can I find information about the SQL Server data types that are supported in the SQL adapter?

To know about the SQL Server data types that are supported in the SQL adapter, see [Basic SQL Server Data Types](#).

Which approach (BizTalk Server, WCF service model or WCF channel model) can I use to perform various operations using the SQL adapter?

To know about the approach that you can use to perform various operations using the SQL adapter, see [Development](#).

Does the Microsoft BizTalk Adapter Pack support tracing?

Microsoft BizTalk Adapter Pack enables adapter clients to activate Windows Communication Foundation (WCF) tracing and adapter-specific tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

Where are the trace files stored?

The trace files are stored at a location that the adapter clients specify. The adapter clients can specify this location while enabling tracing. For more information about tracing, see [Diagnostic Tracing and Message Logging](#).

How do I view the traces?

You can use the WCF Service Trace Viewer tool to view the traces. For more information about the tool, see "Using Service Trace Viewer for Viewing Correlated Traces and Troubleshooting" at <http://go.microsoft.com/fwlink/?LinkId=91243>.

Are performance counters available for adapters?

The BizTalk Adapter Pack provides an "LOB Time (Cumulative)" performance counter to measure the time, in milliseconds, that the LOB client library takes to complete an action initiated by the adapter. For more information about performance counters, see [Using Performance Counters](#).

How do I enable performance counters for adapters?

You can enable or disable the performance counters by setting the **EnablePerformanceCounters** binding property. To enable performance counters, set the **EnablePerformanceCounters** binding property to True. To disable performance counters, set **EnablePerformanceCounters** to False. By default, **EnablePerformanceCounters** is set to False.

See Also

Other Resources

[Frequently Asked Questions](#)

BizTalk Server

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for SQL Server and the BizTalk Server.

Which BizTalk Server tools are used while working with SQL adapter? Where can I know more about these tools?

You should have a working knowledge of the following BizTalk Server tools: BizTalk Explorer, Orchestration Designer, Pipeline Designer, BizTalk Mapper, and BizTalk Server Administration console. To know more about these tools, see [Learn About BizTalk Server Tools](#).

Can I reuse bindings of a BizTalk application? How?

A binding creates a mapping between a logical endpoint such as an orchestration port or a role link and a physical endpoint such as a send and receive port. This enables communication between different components of a BizTalk business solution. The binding information is stored in an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. You can export the bindings of a BizTalk assembly, application, or group, and then reuse it by importing into any other BizTalk application or group. For more information, see [Reusing Adapter Bindings](#).

What is Transaction Isolation Level? How to configure it?

The transaction isolation level determines the degree to which a transaction is isolated from the data changes made by other transactions. It defines the locking behavior of the Transact-SQL commands issued by a connection to SQL Server. For more information about the transaction isolation level, see <http://go.microsoft.com/fwlink/?LinkId=126985>.

For instructions about configuring transaction isolation level in BizTalk Server, see [Configure Transaction Isolation Level and Transaction Timeout](#).

See Also

Other Resources

[Frequently Asked Questions](#)

WCF Service Model

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for SQL Server and the Windows Communication Foundation (WCF) service model.

What is the WCF service model?

The WCF service model is a programming model provided by WCF in which the LOB system (SQL Server database) is exposed as a WCF service. The service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The WCF service model generates a proxy class—the WCF client class—through which your code can invoke operations and receive data using the SQL adapter. For more information about the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

When would I want to use the WCF service model rather than the WCF channel model?

The WCF service model presents a model that is familiar to .NET programmers and that hides the underlying complexities of SOAP message exchange over a channel. Moreover, the Add Adapter Service Reference Plug-in is integrated with the Visual Studio design experience, and presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations exposed by the adapter. Therefore, the WCF service model is often the best choice to develop programming solutions for the SQL adapter.

How do I get started with the WCF service model?

You can use either of the following tools provided by the WCF service model to generate a WCF client class or a WCF service contract and associated helper code from the service metadata that the SQL adapter exposes:

- The ServiceModel Metadata Utility Tool (svcutil.exe), which ships with WCF.
- The Add Adapter Service Reference Visual Studio Plug-in, which ships with the SQL adapter.

For more information about how to generate a WCF client or a WCF service contract, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

See Also

Other Resources

[Frequently Asked Questions](#)

WCF Channel Model

The following are some frequently asked questions (FAQs) related to Microsoft BizTalk Adapter for SQL Server and the Windows Communication Foundation (WCF) channel model.

What is the WCF channel model?

The WCF channel model is a low-level abstraction of the SOAP message exchange between clients and services. It provides interfaces and types that enable you to send and receive messages by using a layered protocol stack called a channel stack. Each layer of the stack is composed of a channel, and each channel is created from a WCF binding. The SQL adapter is a WCF custom transport binding that exposes a SQL database as a WCF service. For more information about the WCF channel model, see [Overview of Using the WCF Channel Model with the Adapter](#).

Why would I want to use the WCF channel model rather than the WCF service model?

You would want to use the WCF channel model over the WCF service model because:

- The WCF channel model provides more fine-grained control over the operations that you perform on the SQL Server database because in the WCF channel model, you directly control the contents of the messages that you send over the channel.
- The WCF channel model provided more comprehensive support for end-to-end streaming of SQL large object (LOB) data types than the WCF service model. This is because in the WCF channel model, you directly control how you provide the message body on outgoing messages and how you process the message body on incoming messages.

How do I get started with the WCF channel model?

Using the WCF channel model, you can invoke operations and receive the results of a polling query by exchanging SOAP messages with the Microsoft BizTalk Adapter for SQL Server over a WCF channel. To get started, you need to create outbound (client) and inbound (service) channels. For more information about creating channels in the WCF channel model, see [Creating a Channel](#).

See Also

Other Resources

[Frequently Asked Questions](#)

Planning and Architecture

This section describes:

- The architecture of end-to-end solutions that use the SQL adapter to operate on a SQL Server database.
- The internal architecture of the SQL adapter.

Understanding the architecture will help you:

- Understand the relationship between the SQL adapter and the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK.
- Understand security boundaries, so that you can improve data security in your solution.
- Understand the SQL adapter binding properties.
- Troubleshoot installation issues.

In This Section

- [Architecture of BizTalk Adapter for SQL Server](#)
- [SQL Adapter and the WCF LOB Adapter SDK](#)

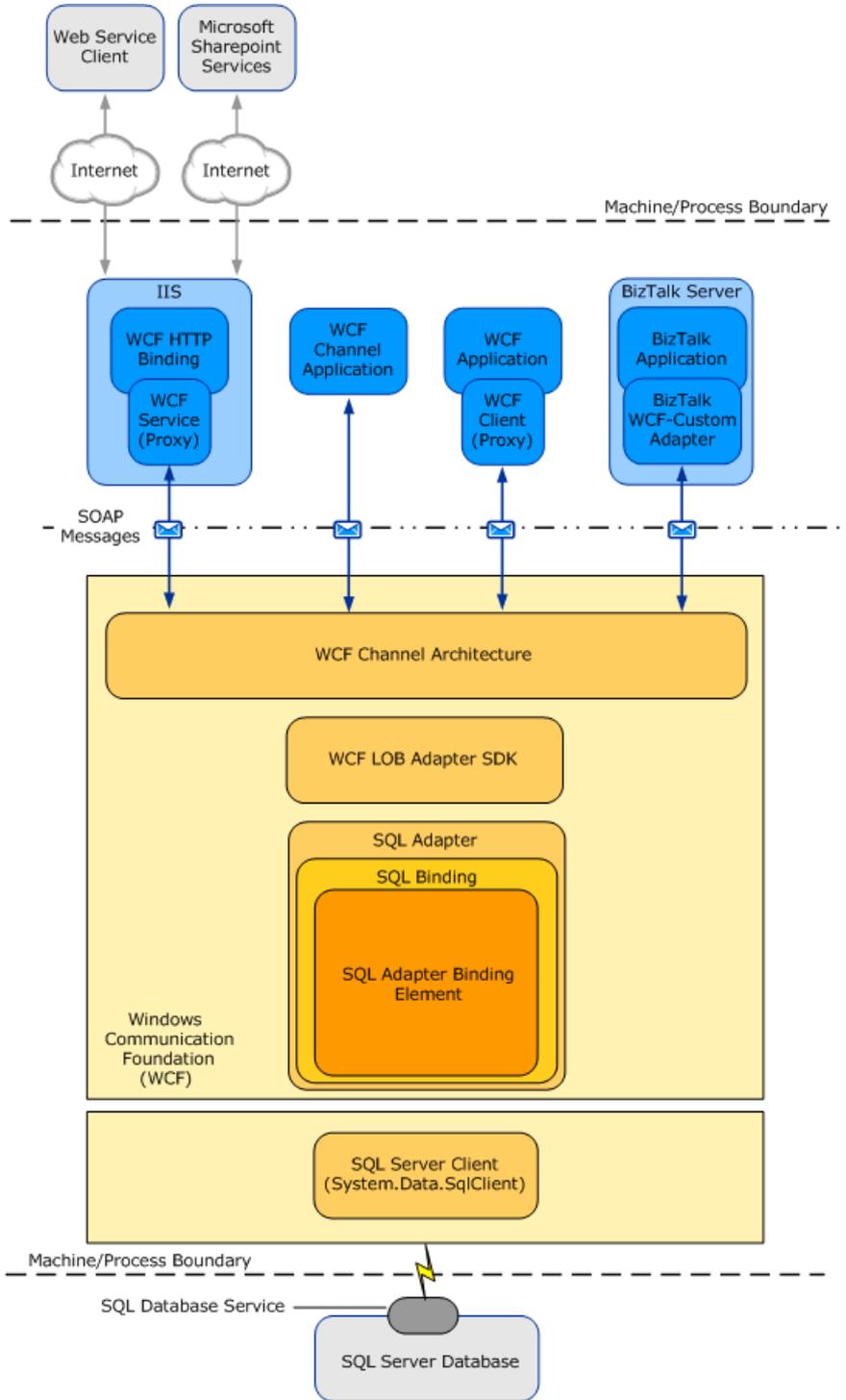
Reference

[SQL Server Developer Center](#)

Architecture of BizTalk Adapter for SQL Server

The Microsoft BizTalk Adapter for SQL Server is a Windows Communication Foundation (WCF) custom binding. This binding contains a single custom transport binding element that enables communication with a SQL Server database. The SQL adapter is wrapped by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK run time and is exposed to applications through the WCF channel architecture. The SQL adapter communicates with the SQL Server database through ADO.NET.

The following figure shows the end-to-end architecture for solutions that are developed by using the SQL adapter.



Consuming the Adapter

The SQL adapter exposes the SQL Server database as a WCF service to client applications. To perform operations and access data on the SQL Server database, client applications exchange SOAP messages with the SQL adapter through WCF channels. The preceding figure shows four ways in which the SQL adapter can be consumed.

- **Through a WCF channel model application.** A WCF channel model application performs operations on the SQL Server database by using the WCF channel model to exchange SOAP messages directly with the SQL adapter.

- **Through a WCF service model application.** A WCF service model application calls methods on a WCF client to perform operations on the SQL Server database. A WCF client models the operations exposed by the SQL adapter as .NET methods. You can use the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK or the WCF ServiceModel Metadata Utility Tool (svcutil.exe) to create a WCF client class from metadata exposed by the SQL adapter.
- **Through a BizTalk receive location or send port that is configured to use the Microsoft BizTalk WCF-Custom adapter.** The WCF-Custom adapter enables the use of WCF extensibility features. By using the WCF-Custom adapter you can select and configure the SQL DB Binding and the behavior for the receive location or send port. For more information about how to use the SQL adapter in BizTalk Server solutions, see [Developing BizTalk Applications](#).
- **Through an IIS-hosted Web service.** In this scenario, a WCF service proxy generated by using the adapter is hosted in IIS using the standard WCF Http Binding. This exposes the service contract as a Web service to external users. IIS automatically hosts the adapter at run time, which, in turn, communicates with the SQL Server database.

The SQL Adapter and WCF

WCF presents a programming model based on the exchange of SOAP messages over channels between clients and services. These messages are sent between endpoints exposed by a communicating client and service. An endpoint consists of:

- An *endpoint address*, which specifies the location at which messages are received.
- A *binding*, which specifies the communication protocols used to exchange messages.
- A *contract*, which specifies the operations and data types exposed by the endpoint.

A binding consists of one or more binding elements that stack on top of each other to define how messages are exchanged with the endpoint. At a minimum, a binding must specify the transport and encoding that are used to exchange messages with the endpoint. Message exchange between endpoints occurs over a channel stack that is composed of one or more channels. Each channel is a concrete implementation of one of the binding elements in the binding configured for the endpoint. For more information about WCF and the WCF programming model, see the WCF Help documentation at <http://go.microsoft.com/fwlink/?LinkId=196855>.

The Microsoft BizTalk Adapter for SQL Server exposes a WCF custom binding, the SQL DB Binding (**Microsoft.Adapters.SQLDB.SQLDBBinding**). By default, this binding contains a single custom transport binding element, the SQL DB Adapter Binding Element (**Microsoft.Adapters.SQLDB.SQLDBAdapter**), which enables operations on a SQL Server database.

Microsoft.Adapters.SQLDB.SQLDBBinding (the SQL DB Binding) and **Microsoft.Adapters.SQLDB.SQLDBAdapter** (the SQL DB Adapter Binding Element) are public classes and are also exposed to the configuration system. Because the SQL DB Adapter Binding Element is exposed publicly, you can build your own custom WCF bindings capable of extending the functionality of the SQL adapter. For example, you could implement a custom binding to support Enterprise Single Sign-on (SSO) in a WCF channel or service model solution. The reasons for doing this would be to aggregate database operations into a single multifunction operation or to perform schema transformation between operations implemented by a custom application and operations on the SQL Server database.

The SQL adapter is built on top of the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, and runs on top of the WCF LOB Adapter SDK run time. The WCF LOB Adapter SDK provides a software framework and tooling infrastructure that the SQL adapter uses to provide a rich set of features to users and adapter clients.

ADO.NET

The SQL adapter connects with the SQL Server database through ADO.NET. ADO.NET provides consistent access to data sources such as SQL Server, and facilitates retrieval, handling, and modification of the data in the data sources. For information about ADO.NET, see <http://go.microsoft.com/fwlink/?LinkId=147356>.

The SQL client provides connectivity to the SQL Server database. You establish a connection to a SQL Server database by providing a connection URI to the SQL adapter. This connection URI contains the name of the computer on which the SQL Server is installed and the name of the database. For more information about the connection URI, see [Establishing a Connection to SQL Server](#).

See Also

Other Resources

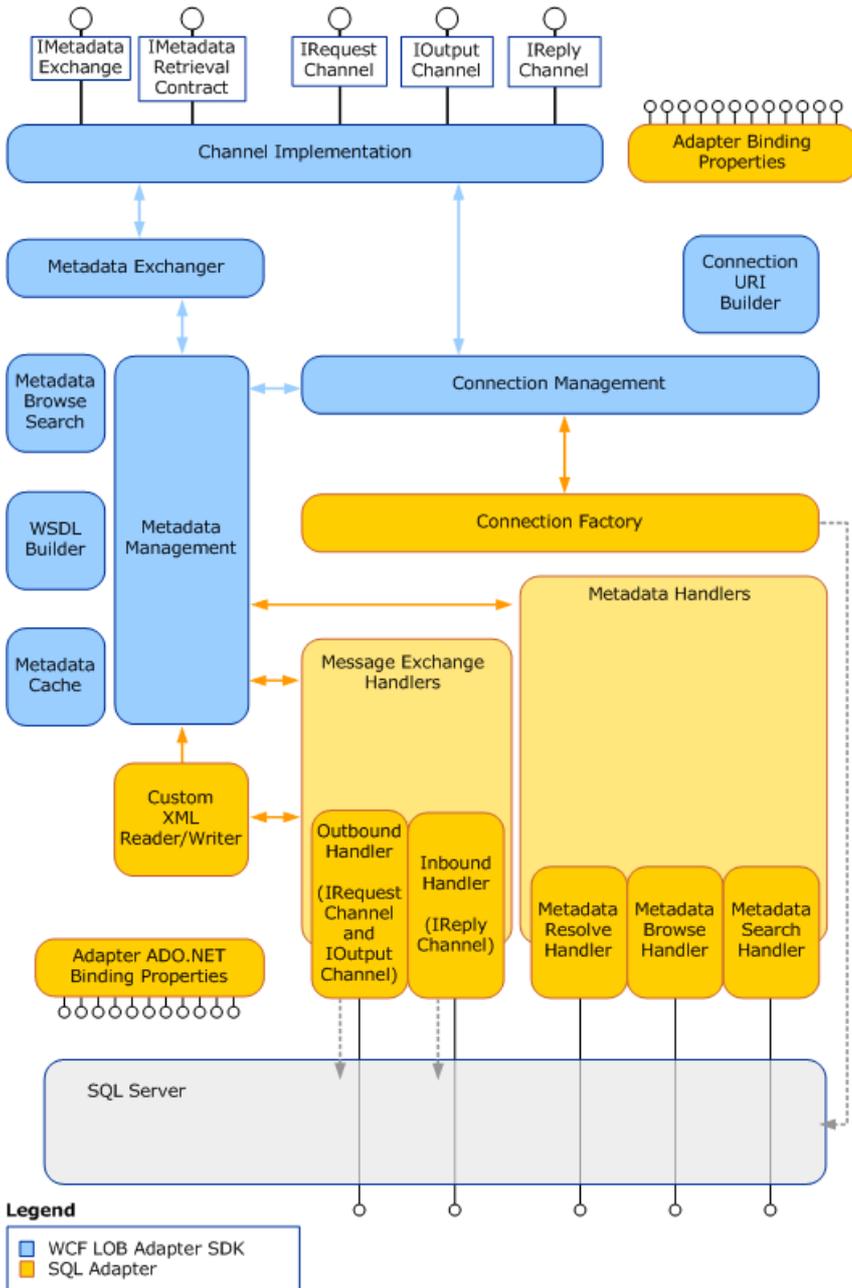
[Planning and Architecture](#)

[Understanding BizTalk Adapter for SQL Server](#)

SQL Adapter and the WCF LOB Adapter SDK

The Microsoft BizTalk Adapter for SQL Server implements a set of core components that leverage functionality provided by the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK and provide connectivity to the SQL Server database through ADO.NET.

The WCF LOB Adapter SDK serves as the software layer through which the SQL adapter interfaces with the Windows Communication Foundation (WCF); ADO.NET serves as the layer through which the SQL adapter interfaces with the SQL Server database. The following figure shows the relationships between the internal components of the SQL adapter and between these components and ADO.NET.



See Also

Other Resources

[Planning and Architecture](#)

Security and Protection

SQL Server databases often contain sensitive business information such as customer account details. Applications that use the Microsoft BizTalk Adapter for SQL Server to access and modify this information either locally or across a distributed network might inadvertently expose it to access by unauthorized actors, unless efforts are made to protect and secure the data during transmission. Data protection and security are usually thought of in the following terms:

- *Authorization* controls access to a resource based on the identity of the requestor.
- *Authentication* provides mechanisms for verifying the identity of a requestor.
- *Data confidentiality* provides mechanisms for protecting the privacy of data through encryption.
- *Data integrity* provides mechanisms to digitally sign data, so that the receiver can ensure that the data has not been altered in-transit.

Another important area of concern is the user-name password credentials that you supply to the SQL adapter. The adapter uses these credentials to open connections to the SQL system. The SQL adapter does not allow credentials to be supplied in the connection URI. This prevents the credentials from getting exposed inadvertently. The SQL adapter provides two alternative methods to supply these credentials in a more secure manner:

Integrated Security. In this case, the SQL adapter uses the Microsoft BizTalk Adapter Pack credentials. You must configure the SQL server to accept these credentials for this method to work.

Enterprise Single Sign-on (SSO). For more information about using SSO, see [Security Considerations When Using the Adapter with BizTalk Server](#).

The topics in this section provide guidelines to help you better secure the solutions that you develop with the SQL adapter.

In This Section

- [Security Considerations Between the SQL Server and Adapter](#)
- [Security Considerations When Using the Adapter with BizTalk Server](#)
- [Security Considerations When Programming on the Adapter](#)
- [Best Practices](#)

Security Considerations Between the SQL Server and Adapter

The SQL adapter is compatible with the standard methods, such as SSO and IPSEC used to secure data exchanges with the database server. Unsecured data exchanges can expose data to unauthorized actors. For information about security issues with SQL Server, see the MSDN article [Security Considerations for SQL Server](#).

You can improve the security of data exchanges by using Internet Protocol Security (IPsec). IPsec is a framework of open standards for protecting communications over Internet Protocol (IP) networks. With IPsec, any data exchanged between the SQL adapter and the SQL server over the network is encrypted, making it difficult for unauthorized actors to use the data. For more information about IPsec and about using IPsec with Microsoft products, see the Microsoft TechNet article [IPsec](#).

The SQL adapter supports SSO and Integrated Security for authentication on the connections that it establishes with the database. With SSO, the credentials are encrypted and stored in the registry. The system uses these credentials to determine access instead of requiring the user to enter them where they might be seen by unauthorized actors. Integrated Security uses the credentials of the logged on user to access the SQL server. This also eliminates the need for users to enter credentials. The database administrator must configure SQL to accept the users credentials for Integrated Security to work correctly.

See Also

Other Resources

[Security and Protection](#)

Security Considerations When Using the Adapter with BizTalk Server

When you configure a send port or a receive port (location) by using the BizTalk Server Administration console or use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas for a BizTalk solution, you must provide credentials for the SQL Server database. It is important to supply these credentials in a secure way to help prevent them from being revealed to potentially malicious actors. This topic discusses how to most securely supply credentials for the Microsoft BizTalk Adapter for SQL Server for BizTalk Server solutions.

A more general discussion of security in the context of BizTalk solutions is an expansive topic and is beyond the scope of this documentation. For information about how you can make your BizTalk solutions more secure, see "Security and Protection" at <http://go.microsoft.com/fwlink/?LinkID=101135> in the BizTalk Server Help documentation.

How Do I Protect Credentials When I Use the Consume Adapter Service BizTalk Project Add-in?

When you use the Consume Adapter Service Add-in to retrieve message schemas for a BizTalk solution, you must supply the user name and password from the **Security** tab on the **Configure Adapter** dialog box. The Consume Adapter Service Add-in will not allow you to set credentials in the **Configure a URI** field. This improves security by preventing credentials from appearing in clear text. For more information about how to retrieve message schemas by using the Consume Adapter Service Add-in, including how to enter a user name and password for the SQL Server database, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#).

How Do I Protect Credentials When I Configure a Send Port or a Receive Location?

BizTalk solutions use the Microsoft BizTalk WCF-Custom adapter to consume WCF services. The SQL adapter is a WCF custom binding that enables clients to consume the SQL Server database as if it were a WCF service. BizTalk solutions consume the SQL adapter through send ports and receive locations that are configured to use the WCF-Custom adapter. The WCF-Custom adapter is, in turn, configured to use the SQL adapter as its transport. For more information about how to configure send ports and receive ports (receive locations), including how to configure the WCF-Custom adapter, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

You configure the SQL Server database credentials from the **Credentials** tab of the **WCF-Custom Transport Properties** dialog box for send ports or from the **Other** tab of the **WCF-Custom Transport Properties** dialog box for receive locations. Because the WCF-Custom adapter supports Enterprise Single Sign-On (SSO), you can also choose to provide either a user name and password or an SSO affiliate application on either of these tabs. The following topics discuss both options.

User Name Password Credentials

You should only supply a user name and password from the **Credentials** tab (for send ports) or the **Other** tab (for receive locations) in the **WCF-Custom Transport Properties** dialog box. This ensures the following:

- Your credentials will not be displayed in the **Address (URI)** field of the dialog box. This prevents those who have access to your screen (or who have permissions that enable them to view the send port or receive location properties) from seeing your credentials.
- Your password will not be written to the binding file if you export the send port or receive port binding. This prevents anyone with access to the file from viewing your password.

Enterprise Single Sign-On and SSO Affiliate Applications

You can configure the WCF-Custom adapter so that it uses Enterprise Single Sign-on (SSO) to get the credentials for the SQL Server database. SSO uses a database and a master secret to encrypt and store user credentials. It also provides services to map Microsoft Windows accounts to secondary credentials that are used to access a back-end system. By using SSO, you can map a Windows account to a user name and password on the SQL Server database.

SSO uses *affiliate applications* and *SSO mappings* to map credentials to the back-end system. An affiliate application is a logical entity in SSO that refers to a system or an application that requires secondary credentials. An SSO mapping is associated with an affiliate application. It maps a Windows account to the secondary credentials used by that account to access the affiliate system or application. An SSO mapping can be associated with a Windows user account or with a group.

To use SSO with the SQL adapter, you must do the following.

1. Create an affiliate application in SSO to hold the user name password credentials for the SQL Server database. This step

is often performed by someone with special types of SSO administrative privileges.

2. Create a user or group mapping for the affiliate application that maps your Windows account to the user name and password that are used to establish a connection with the SQL Server database. Depending on your installation, a user might be able to perform this step, or it might require someone with special types of SSO administrative privileges.

 **Note**

When configured for SSO, the WCF-Custom adapter uses services provided by SSO to get the SQL Server user name and password from the SSO database. It provides these (unencrypted) to the SQL adapter, so that the adapter can open a connection to the SQL Server database. SSO provides no encryption or protection across the connection between the SQL adapter and the SQL Server database.

For information about how to use SSO, including information about how to create affiliate applications and SSO mappings, see the "Using SSO" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103052>. For more general information about SSO, see the "Implementing Enterprise Single Sign-On" topic in the BizTalk Server documentation at <http://go.microsoft.com/fwlink/?LinkId=103054>.

The `AcceptCredentialsInUri` Binding Property

The SQL adapter does not support **AcceptCredentialsInUri** binding property. Credentials are never permitted in the connection URI.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Security Considerations When Programming on the Adapter

How Do I Protect Credentials When I Use the Add Adapter Service Reference Visual Studio Plug-in?

When you use the Add Adapter Service Reference Plug-in to create a WCF client, you might have to supply a user name and password for the SQL Server database. You must enter credentials from the **Security** tab on the **Configure Adapter** dialog box. The SQL adapter does not provide an option to specify the user name and password as part of the connection URI. This ensures the following:

- The credentials will not be displayed in the **Configure a URI** field of the **Add Adapter Service Reference Plug-in** dialog box where anyone with access to your computer screen can read them.
- The credentials will not appear in the configuration file that the Add Adapter Service Reference Plug-in generates.

For more information about how to generate a WCF client by using the Add Adapter Service Reference Plug-in, including how to enter a user name and password for the SQL Server database, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#).

What Are Best Practices for Setting Credentials in Code?

WCF provides the **ClientCredentials** class to help you configure the credentials that a client communication object, such as a **ChannelFactory**, uses to authenticate itself with a service. By using the **ClientCredentials** class, you ensure that WCF takes whatever authentication mechanisms are specified in that object's channel stack and applies them to the exchange between your client and the service.

Because the SQL adapter is hosted in-process with its consuming application, it is not imperative to use the **ClientCredentials** class to set credentials on the client communication objects that the consuming application uses. It is, however, considered good practice to do so.

The SQL adapter requires the use of the **ClientCredentials** class for programmatically passing credentials. The **AcceptCredentialsInUri** binding property is ignored by the SQL adapter to prevent passing credentials in the URI.

The following example shows how to use the **Credentials** property to set credentials for the SQL Server database on a **ChannelFactory**.

```
// Create binding and endpoint
SqlAdapterBinding binding = new SqlAdapterBinding();
EndpointAddress address = new EndpointAddress("mssql://mysqlserver//mydatabase?");

// Create the channel factory
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding, endpointAddress);

// Set user name and password
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open the channel factory
factory.Open();
```

The following example shows how to use the **ClientCredentials** class to set credentials for the SQL Server database on a WCF client.

```
// Initialize a new client for the SELECT operation on the Employee table
SqlAdapterBinding binding = new SqlAdapterBinding();
EndpointAddress address = new EndpointAddress("mssql://mysqlserver//mydatabase?");
TableOp_dbo_EmployeeClient client = new TableOp_dbo_EmployeeClient(binding, address);

// Set user name and password
client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";

// Open the client
```

```
client.Open();
```

How Can I Provide for More Secure Data Exchange Across Process Boundaries?

The SQL adapter is hosted in-process with the application or service that consumes it. Because the adapter is hosted in-process with the consumer, there is no need to provide security on messages exchanged between the consumer and the SQL adapter. However, if the consuming application or service sends messages that contain sensitive database information across a process boundary to another service or client, you should take measures to provide adequate protection for this data in your environment. Windows Communication Foundation (WCF) provides many options for helping to secure messages sent between clients and services. For more information about helping to secure messages sent between clients and services in WCF, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>. For more general information about security features that WCF provides, see "Windows Communication Foundation Security" at <http://go.microsoft.com/fwlink/?LinkId=89726>.

See Also

Concepts

[Best Practices](#)

Other Resources

[Security and Protection](#)

Best Practices

This section provides best practices that you should follow to more completely protect sensitive data when you use or develop applications that consume the Microsoft BizTalk Adapter for SQL Server.

Security Best Practices for the Connection between the SQL Adapter and the SQL Server Database

- The SQL adapter provides no support for helping to secure communication between it and the SQL Server database. You must provide a mechanism to help ensure an adequate level of security for data exchanged between the adapter and the SQL Server database.
- For security reasons, the SQL adapter does not allow you to provide user name password credentials for the SQL Server database in the connection URI. See the remainder of this topic for alternative methods of providing credentials to the SQL adapter.
- The SQL adapter also enables you to use Windows Authentication while connecting to SQL Server to generate metadata and perform operations, either through Visual Studio or BizTalk Server. Before using Windows Authentication, you must add the Windows user as a user in SQL Server Management Studio. For more information, see [Connecting to SQL Server Using Windows Authentication](#).

For more information, see [Security Considerations Between the SQL Server and Adapter](#).

Security Best Practices for Consuming the SQL Adapter with BizTalk Server

- The SQL adapter does not allow you to provide user name password credentials for the SQL Server database in the connection URI.
- When you use the Consume Adapter Service Add-in, enter the user name password credential for the SQL Server database from the **Security** tab of the **Configure Adapter** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the SQL adapter on a send port, enter the user name password credential for the SQL Server database from the **Credentials** tab of the **WCF-Custom Transport Properties** dialog box.
- When you configure the BizTalk WCF-Custom adapter for the SQL adapter on a receive location, enter the user name password credential for the SQL Server database from the **Other** tab of the **WCF-Custom Transport Properties** dialog box.
- While using Consume Adapter Service Add-in to generate metadata, configuring send port, or configuring receive port, you can also use Windows Authentication. Before using Windows Authentication, you must add the Windows user as a user in SQL Server Management Studio. For more information, see [Connecting to SQL Server Using Windows Authentication](#).

For more information, see [Security Considerations When Using the Adapter with BizTalk Server](#).

Security Best Practices for Consuming the SQL Adapter with Programming Solutions

- It is sometimes necessary to provide the user name password credentials for the SQL Server database in the connection URI; however, if possible, you should avoid doing this.
- When you use the Add Adapter Service Reference Visual Studio Plug-in, enter the user name password credential for the SQL Server database from the **Security** tab of the **Configure Adapter** dialog box.
- In WCF channel model programming, use the **Credentials** property on the channel factory to set the user name password credential for the SQL Server database.
- In WCF service model programming, use the **ClientCredentials** property on the WCF client to set the user name password credential for the SQL Server database.

- If an application that consumes the SQL adapter sends messages that contain sensitive database information across a process boundary to another service or client, ensure that these messages have sufficient security measures applied to provide adequate data protection in your environment.
- While using Add Adapter Service Reference Plug-in or connecting to SQL Server from a .NET application, you can also use Windows Authentication. Before using Windows Authentication, you must add the Windows user as a user in SQL Server Management Studio. For more information, see [Connecting to SQL Server Using Windows Authentication](#).

For more information, see [Security Considerations When Programming on the Adapter](#).

Security Best Practices for Hosting the SQL Adapter in IIS

Hosting the SQL adapter in Microsoft Internet Information Services (IIS) as a Web service exposes operations surfaced by the SQL adapter to Web clients. These operations might involve exchanging sensitive data over the Internet, so you should take measures to help ensure that this data is as secure as possible.

WCF provides two standard bindings for HTTP transport: the **BasicHttpBinding** provides basic HTTP transport with no security mechanisms; the **WSHttpBinding** supports both transport-level and message-level security mechanisms.

You can either use the **BasicHttpBinding** over an HTTPS connection, or use the **WSHttpBinding** to help protect your data. The WCF LOB Adapter SDK includes the WCF LOB Adapter Service Development Wizard to generate WCF service for LOB artifacts. This wizard only supports use of **BasicHttpBinding**.

You can also develop a custom HTTP binding to leverage additional security mechanisms that your environment provides. For more information about the security features that WCF provides, see "Securing Services and Clients" at <http://go.microsoft.com/fwlink/?LinkId=89725>.

When hosting the SQL adapter as a Web service, Web developers should take measures to prevent strings typed in by users from being passed directly to the SQL Server database. For example, if a Web site lets the user enter a value that will be part of a WHERE clause in a SELECT statement, the input string should be scanned to prevent adding other commands to the statement.

Security Best Practices for WCF Diagnostic Tracing and Message Logging

WCF supports diagnostic tracing and message logging. You configure diagnostic tracing and message logging either through configuration files or by using Windows Management Instrumentation (WMI). Depending on the configuration options you set, WCF diagnostic tracing or message logging can emit sensitive information to log files, where it could potentially be exposed to observation by unauthorized users.

Follow the recommendations provided in the WCF documentation to mitigate potential security threats exposed by enabling these features. At a minimum, you should observe the following best practices for diagnostic tracing and message logging:

- Do not enable "verbose" or "information" tracing in a production environment. This may lead to performance degradation. However, you must enable "warning" and "error" tracing in a production environment. If you enable tracing, you must take proper security measures to protect your data. See the WCF documentation for more information.
- Ensure that log files and configuration files are protected by access control lists (ACLs).

The following warnings apply specifically to the messages that are exchanged between a client application and the SQL adapter:

- WCF diagnostic tracing can log the header (but not the body) of messages exchanged with the SQL adapter. Because the message action is in the message header, this reveals the operations invoked on the SQL adapter by the client.
- If WCF message logging is enabled and **logMessagesAtServiceLevel** is **true**, the message header (but not the message body) of messages exchanged between the adapter client and the SQL adapter are logged. Because the message action is in the message header, this reveals the operations that the client invoked on the SQL adapter. If **logEntireMessage** is also **true**, the message body will be logged. This can reveal sensitive database information.

For more information about improving security when you enable diagnostic tracing, see "Security Concerns and Useful Tips for Tracing" at <http://go.microsoft.com/fwlink/?LinkId=89796>. For more information about improving security when you enable

message logging, see "Security Concerns for Message Logging" at <http://go.microsoft.com/fwlink/?LinkId=89797>.

See Also

Other Resources

[Security and Protection](#)

Development

The Microsoft BizTalk Adapter for SQL Server is a Windows Communication Foundation (WCF) binding. Client applications can consume the SQL adapter to invoke operations on SQL Server artifacts. The SQL adapter can be consumed:

- Through a physical port binding in a BizTalk Server solution.
- By invoking methods on an instance of a client proxy.
- As a hosted WCF service.
- By sending SOAP messages over a channel instance in code that uses the WCF channel model.

The following table:

- Lists the different operations that can be performed on SQL Server using the SQL adapter.
- Provides links to the topics containing information about performing the task using the chosen approach (BizTalk Server, WCF service model, WCF channel model).

T a s k	BizTalk Server	WCF Service Model
P e r f o r m i n g b a s i c I n s e r t, U p d a t e, D e l e t e, a n d S e l e c t o p e r a t i o n s o n t a b l e s a n d v i e w s	Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server	Performing Insert, Update, Delete, and Select Operations by Using the WCF Service Model

Performing Operations on Tables and Views with Large Data Types by Using BizTalk Server

Performing Operations on Tables and Views with Large Data Types by Using the WCF Service Model

Per
for
r
m
in
g
o
p
er
at
io
n
s
o
n
ta
bl
e
s
a
n
d
vi
e
w
s
w
it
h
la
r
g
e
d
at
a
ty
p
e
c
ol
u
m
n
s
(
Al
s
o
in
cl
u
d
e
s
in
fo
r
m
at
io
n
a
b
o
ut
FL
L
E
S
T
R
E
A
M
o
p
er
at
io
n
s
u
si
n
g
th
e
a
d
a
pt
er
)

<p>Executing Stored Procedures in SQL Server by Using BizTalk Server</p>	<p>Invoking Stored Procedures Using the WCF Service Model</p>
<p>Executing Stored Procedures With a Single XML Parameter</p>	

E x e c u t i n g s t o r e d p r o c e d u r e s t h a t c o n t a i n a F O R X M L c l a u s e i n t h e d e f i n i t i o n	Executing Stored Procedures Having a FOR XML Clause	
P e r f o r m i n g c o m p o s i t e o p e r a t i o n s o n S Q L S e r v e r	Performing Composite Operations on SQL Server by Using BizTalk Server	

Invoking Scalar Functions in SQL Server by Using BizTalk Server	Invoking Scalar Functions in SQL Server by Using the WCF Service Model
Invoking Table-Valued Functions in SQL Server by Using BizTalk Server	Invoking Table-Valued Functions in SQL Server by Using the WCF Service Model

<p>Per for m in g E x e c u t e R e a d e r, E x e c u t e S c a l a r, o r E x e c u t e N o n Q u e r y o p e r a t i o n s</p>	<p>Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using BizTalk Server</p> <p>Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using WCF Service Model</p>
<p>R e c e i v i n g p o l l i n g - b a s e d d a t a- c h a n g e m e s s a g e s</p>	<p>Polling SQL Server by Using the SQL Adapter with BizTalk Server</p> <p>Polling SQL Server by Using the SQL Adapter with WCF Service Model</p>

Receiving Query Notifications by Using BizTalk Server	Receiving Query Notifications Using the WCF Service Model
---	---

The topics in this section provide information, procedures, and examples to help you develop applications that consume the SQL adapter in both BizTalk Server and .NET programming solutions. The topics also provide information on other key aspects of using the adapters such as:

- [Connecting to SQL Server.](#)
- [Retrieving metadata from SQL Server.](#)
- [Using binding properties to configure the adapter.](#)

In This Section

- [Establishing a Connection to SQL Server](#)
- [Retrieving Metadata for SQL Server Operations in Visual Studio](#)
- [Working with BizTalk Adapter for SQL Server Binding Properties](#)
- [Before Developing Applications Using the SQL Adapter](#)
- [Developing BizTalk Applications](#)
- [Developing Applications by Using the WCF Service Model](#)
- [Developing Applications by Using the WCF Channel Model](#)
- [Samples](#)
- [How Do I?](#)

Establishing a Connection to SQL Server

The Microsoft BizTalk Adapter for SQL Server is a Windows Communication Foundation (WCF) custom binding. As such, it enables communication to a SQL Server database through a WCF endpoint address. In WCF, the endpoint address identifies the network location of a service and is typically expressed as a Uniform Resource Identifier (URI). The SQL adapter expresses this location as a connection URI, which contains properties that the SQL adapter uses to establish a connection to the SQL Server database. You must specify a connection URI when you:

- Create a channel factory or a channel listener using the WCF channel model or when you create a WCF client or service host using the WCF service model.
- Create a physical port binding in a BizTalk Server solution.
- Use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class or WCF service interface for a WCF service model solution.
- Use the Consume Adapter Service BizTalk Project Add-in to retrieve message schemas from the SQL adapter for a BizTalk Server solution.
- Use the ServiceModel Metadata Utility tool (svcutil.exe) to generate a WCF client class or WCF service interface for a WCF service model solution.

The topics in this section describe how to establish a connection between the Microsoft BizTalk Adapter for SQL Server and the SQL Server database by providing you with:

- Information about the connection properties and the structure of the SQL Server connection URI.
- Links to topics that show how to specify a connection URI by using the SQL adapter.
- Information about connecting to SQL Server using Windows Authentication.

In This Section

- [SQL Server Connection URI](#)
- [Connecting to SQL Server Using Windows Authentication](#)

See Also

Other Resources
[Development](#)

SQL Server Connection URI

The Microsoft BizTalk Adapter for SQL Server connection URI contains properties that the adapter uses to establish a connection to the SQL Server database. This topic provides information about the SQL Server connection URI, and provides links to other topics that explain how to specify a URI in different programming scenarios.

The Connection URI for the SQL Adapter

A typical endpoint address URI in WCF is represented as: `scheme://hostinfoparams?query_string`, where:

- `scheme` is the scheme name.
- `hostinfoparams` is information required to establish the connection to the host; for example, a server name.
- `query_string` is an optional name-value collection of parameters delimited by a question mark (?).

The SQL adapter connection URI adheres to this basic format and is implemented as follows:

```
mssql://[Server_Name[:Portno]]/[Database_Instance_Name]/[Database_Name]?FailoverPartner=[Partner_Server_Name]&InboundId=[Inbound_ID]
```

where, `mssql` is the scheme for the SQL Server connection URI.

The following table explains the properties contained in the connection URI.

Connection URI Property	Category	Description
[SERVER_NAME]	hostinfoparams	Name of the server on which SQL Server is installed. If you do not specify a value, the adapter assumes the server name as "localhost" and establishes a connection with the SQL Server database on the local server.
[PORTNO]	hostinfoparams	The port number where the connection is established. If you do not specify a value, the adapter connects through the default port.
[DATABASE_INSTANCE_NAME]	hostinfoparams	Name of the SQL Server instance to connect to. If you do not specify a value, the adapter connects to the default database instance.

[DATABASE_NAME]	hostname	Name of the database to connect to. If you do not specify a value, the adapter connects to the default database.
[PARTNER_SERVER_NAME]	querystring	Name of the failover SQL Server database to connect to if the primary SQL Server database is not available. For more information about high availability with respect to SQL Server, see "Using Database Mirroring" at http://go.microsoft.com/fwlink/?LinkID=196852 .
[INBOUND_ID]	querystring	An identifier that you add to the connection URI to make it unique. You must provide this connection parameter if you want to generate metadata for the TypedPolling inbound operation. Also, in a BizTalk application, if you have multiple receive locations polling the same database, the inbound ID makes the connection URI unique, thereby enabling adapter clients to receive polling messages from the same database on different receive locations. For more information, see Receiving Polling Messages Across Multiple Receive Ports .

Note

For more information about these connection string properties, see "SqlConnection.ConnectionString Property" at <http://go.microsoft.com/fwlink/?LinkId=196958>.

SQL Server Credentials and the Connection URI

The SQL adapter does not support specifying credentials in the connection URI. For more information about specifying credentials in your applications that use the SQL adapter, see [Security and Protection](#).

Using Special Characters in the Connection URI

The SQL adapter does not support specifying a connection URI that has special characters for any of the parameter values. If the connection parameter values contain special characters, make sure you do one of the following:

- If you are specifying the URI in Visual Studio using Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. If you specify the URI directly in the **Configure a URI** field and the connection parameters contain special characters, you must specify the connection parameters using proper escape characters.

For example, if the connection URI has a parameter with name *sql server*, you must specify it as *sql%20server*.

- If you are specifying the URI while creating a send or receive port in BizTalk Server Administration console, and the connection parameters contain special characters, you must specify the connection parameters using proper escape characters.

Using the Connection URI to Connect to the SQL Server Database

The following is a sample connection URI for the SQL adapter.

```
mssql://sql_server/sql_server_instance//
```

In the preceding example, "sql_server" is the name of the computer on which SQL Server is installed whereas "sql_server_instance" is the name of the database instance to connect to. Because no database name is specified, the adapter will connect to the default database.

The following is an example of a connection URI where the SQL Server database is installed on the same computer as the SQL adapter. In this example, the adapter connects to the database "my_database" for the "sql_server_instance" database instance

on the local computer.

```
mssql://localhost/sql_server_instance/my_database/
```

In this example, the adapter connects to the default database for the default instance running on the local computer.

```
mssql://localhost///
```

For information about how to specify a connection to the SQL Server database when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to SQL Server in Visual Studio](#).
- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Specifying a Client Binding for the SQL Adapter](#).

See Also

Other Resources

[Development](#)

Connecting to SQL Server Using Windows Authentication

The SQL adapter enables adapter clients to use Windows Authentication to establish a connection with SQL Server. To use Windows Authentication adapter clients must specify an empty user name and password. For more information about connecting to SQL Server using Windows Authentication, see [Connecting to SQL Server in Visual Studio](#).

To enable adapter clients to use Windows Authentication to connect to SQL Server, you must perform the following tasks on the computer running SQL Server.

Note

The steps provided here are for SQL Server 2008 and SQL Server 2005. These steps may vary for SQL Server 2000. Refer to SQL Server documentation for more information.

To enable Windows Authentication on SQL Server

1. Start SQL Server Management Studio and in the **Connect to Server** dialog box, specify a SQL Server name and administrator credentials to connect to the server.
2. In **Object Explorer**, expand the SQL Server.
3. Expand **Security**, right-click **Logins**, and then click **New Login**.
4. In the **Login – New** window, for the **Login name** text box, specify the Windows user name in the format domain\username.
5. In the left pane, click **User Mapping**. In the right pane, select a database to associate with the user. In the **Database role membership for** box, select the **db_owner** check box.
6. Click **OK**.

After you have added the user to SQL Server, the user can connect to SQL Server using the SQL adapter, logging in with a blank username and password.

See Also

Other Resources

[Establishing a Connection to SQL Server](#)

Retrieving Metadata for SQL Server Operations in Visual Studio

The Microsoft BizTalk Adapter for SQL Server provides three Visual Studio components that you can use to help you develop solutions using the adapter—the Consume Adapter Service BizTalk Project Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Visual Studio Plug-in. Adapter clients must use these components to connect to SQL Server and then generate metadata for the operations they want to perform.

All these Visual Studio components simplify development by:

- Providing a Microsoft Windows interface through which you can browse and search for operations that you want to use in your solution.
- Retrieving metadata exposed by the adapter for these target operations.
- Converting that metadata, which is expressed as a Web Services Description Language (WSDL) document by the adapter, into a form that you can use in your solution (XSD message schemas for BizTalk projects or a .NET object representation of a service contract for the WCF service model) and adding it to your project.

This section provides instructions about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

In This Section

- [Connecting to SQL Server in Visual Studio](#)
- [Browsing, Searching, and Retrieving Metadata for SQL Server Operations](#)

Connecting to SQL Server in Visual Studio

This section provides instructions about how to use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in.

- The **Consume Adapter Service BizTalk Project Add-in** is available in BizTalk Server projects and is installed as part of the WCF LOB Adapter SDK installation. You use the Consume Adapter Service Add-in to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).
- The **Add Adapter Metadata Wizard** is available in the BizTalk Server projects and is installed as part of the BizTalk Server installation. You use the Add Adapter Metadata Wizard to generate message schemas (XSDs) for operations that you want to target in your BizTalk solution. For more information about developing solutions with BizTalk Server, see [Developing BizTalk Applications](#).

Note

Because the SQL adapter is exposed both as a WCF-Custom binding and as BizTalk adapter, you can use either the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard from a BizTalk project to connect to SQL Server.

- The **Add Adapter Service Reference Visual Studio Plug-in** is available in non-BizTalk programming projects. You use the Add Adapter Service Reference Plug-in to generate a WCF client class or a WCF service callback interface when you develop solutions using the WCF service model. For more information about developing solutions with the WCF service model, see [Developing Applications by Using the WCF Service Model](#).

To use the Consume Adapter Service BizTalk Project Add-in, Add Adapter Metadata Wizard, or the Add Adapter Service Reference Visual Studio Plug-in, you must first connect to SQL Server. All three methods present a dialog box through which you configure a connection by setting the following:

- **Connection parameters.** These are the parameters that are used to build the connection URI such as the SQL Server name, the database instance name, and the database name.
- **User name password credentials for SQL Server.** These are used to authenticate you on SQL Server when the connection is established. You must specify a user name and password.
- **Binding properties.** Binding properties are optional, and whether you specify them depends primarily on whether you target operations that require specific binding properties to be set. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

At a minimum, when you configure the connection to SQL Server, you only have to specify binding properties and connection parameters that are needed to establish the connection and that affect the metadata returned by the SQL adapter for the operations you want to target. However, you might also want to specify values for any additional binding properties and connection parameters that will be used at run time. This is because:

- The Consume Adapter Service Add-in creates a BizTalk port binding file from the binding properties and connection parameters that you specify when you configure the connection and adds this file to your project.
- The Add Adapter Service Reference Plug-in creates an app.config file from the binding properties and connection properties that you specify when you configure the connection and adds this file in your project directory.

In This Section

- [Connecting to SQL Server in Visual Studio Using Consume Adapter Service Add-in](#)
- [Connecting to SQL Server in Visual Studio Using Add Adapter Metadata Wizard](#)

- [Connecting to SQL Server in Visual Studio Using Add Adapter Service Reference Plug-in](#)

Connecting to SQL Server in Visual Studio Using Consume Adapter Service Add-in

The Consume Adapter Service BizTalk Project Add-in is installed when you install WCF LOB Adapter SDK. The Consume Adapter Service Add-in loads all the WCF-Custom bindings installed on the computer. To connect to SQL Server using the WCF-based SQL adapter in a BizTalk project, you must use the **sqlbinding**.

This topic provides instructions on how to use the Consume Adapter Service Add-in.

Connecting to SQL Server Using Consume Adapter Service Add-in

Perform the following steps to connect to SQL Server using the Consume Adapter Service Add-in.

To connect to SQL Server

1. To connect using the Consume Adapter Service Add-in in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project name in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

- d. Click **Add**. The Consume Adapter Service Add-in opens.
2. From the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
 3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list, do one of the following:

 Note
If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication .

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
User name	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive.  Note If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.

4. Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).

 Note

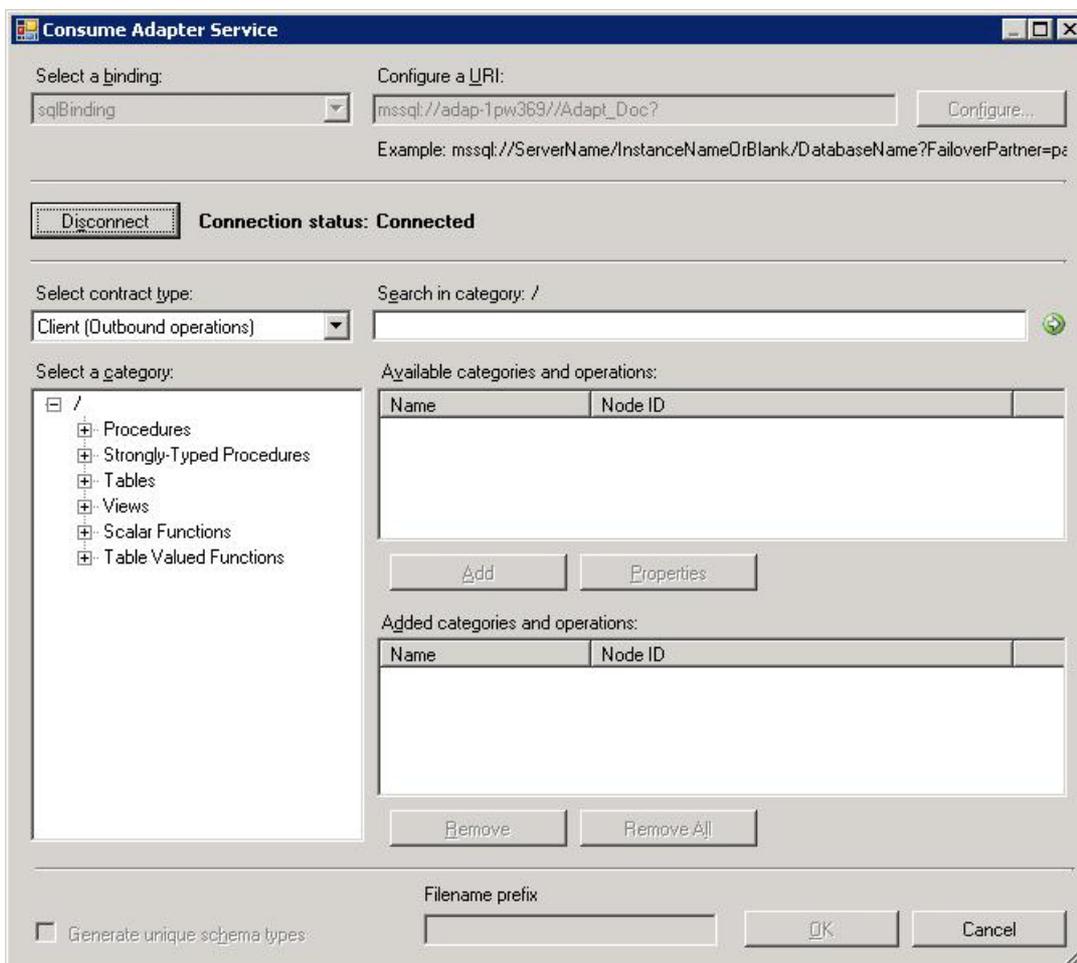
If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

Note

If you do not specify any values in the URI property tab, the Consume Adapter Service Add-in puts the URI as `mssql://.//.` In such a case, the adapter connects to the default database and the default database instance on the local computer.

5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target.
6. Click **OK**.
7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on SQL Server. For example, the **Procedures** node contains all the procedures available for the database you connected to. Similarly, the **Tables** node contains all the tables in the database you connected to, and the operations that can be performed on a table. For more information about these nodes, see [Metadata Node IDs](#).

See Also

Other Resources

[Connecting to SQL Server in Visual Studio](#)

Connecting to SQL Server in Visual Studio Using Add Adapter Metadata Wizard

The SQL adapter is also exposed as a BizTalk adapter and, therefore, you can use the Add Adapter Metadata Wizard to generate schema for the operations you want to perform on SQL Server using the adapter.

Connecting to SQL Server Using Add Adapter Metadata Wizard

Perform the following steps to connect to SQL Server using the Add Adapter Metadata Wizard.

To connect to SQL Server

1. To connect using the Add Adapter Metadata Wizard in a BizTalk solution:
 - a. Create a BizTalk project using Visual Studio.
 - b. Right-click the project name in Solution Explorer, point to **Add**, and then click **Add Generated Items**.
 - c. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

- d. Click **Add**. The Add Adapter Metadata Wizard opens.
- e. In the Add Adapter Wizard, select **WCF-SQL**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-SQL port configured in BizTalk, select the port from the Port list.

- f. Click **Next**.
2. From the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list, do one of the following:

📌 Note
If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication .

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.

User name	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive.  Note If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.
------------------	---

- Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).

 Note If the connection parameters contain any reserved characters, you must specify them as-is in the URI Properties tab, that is, without using any escape characters. However, if you specify the URI directly in the Configure a URI field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

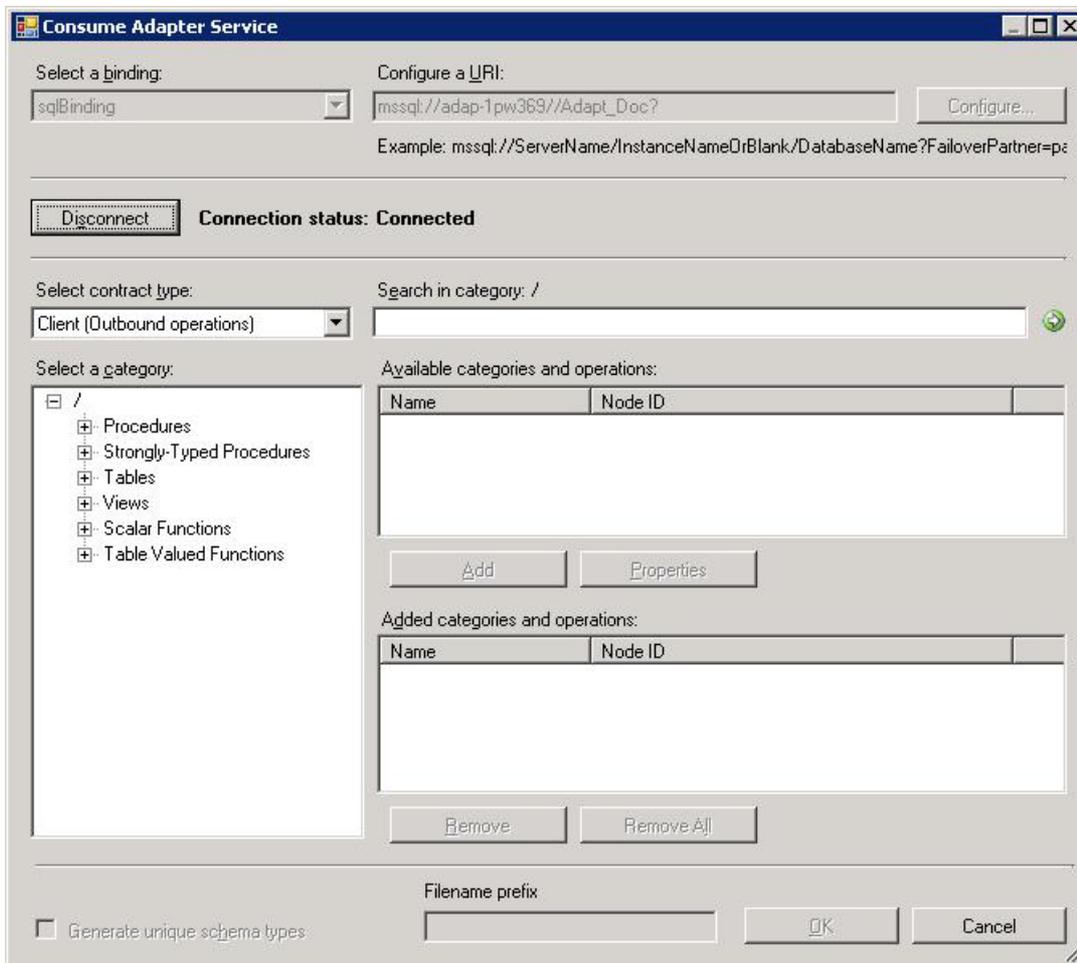
 Note If you do not specify any values in the URI property tab, the Consume Adapter Service Add-in puts the URI as <code>mssql://.//.</code> In such a case, the adapter connects to the default database and the default database instance on the local computer.

- Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties see, [Working with BizTalk Adapter for SQL Server Binding Properties](#).

 Note If you are generating metadata using Add Adapter Metadata Wizard and you selected an existing WCF-SQL send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

- Click **OK**.
- Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established. The graphical user interface is same for the Add Adapter Metadata Wizard.



The Consume Adapter Service Add-in displays different nodes containing various operations that can be performed on SQL Server. For example, the **Procedures** node contains all the procedures available for the database you connected to. Similarly, the **Tables** node contains all the tables in the database you connected to, and the operations that can be performed on a table. For more information about these nodes, see [Metadata Node IDs](#).

See Also

Other Resources

[Connecting to SQL Server in Visual Studio](#)

Connecting to SQL Server in Visual Studio Using Add Adapter Service Reference Plug-in

To connect to SQL Server using the SQL adapter in a .NET programming solution, you must use the Add Adapter Service Reference Visual Studio Plug-in. This topic provides instructions on how to use the Add Adapter Service Reference Plug-in.

Connecting to SQL Server Using Add Adapter Service Reference Plug-in

Perform the following steps to connect to SQL Server using the Add Adapter Service Reference Plug-in.

To connect to SQL Server

1. To connect using the Add Adapter Service Reference Plug-in in a programming solution:
 - a. Create a project in Visual Studio.
 - b. Right-click the project in Solution Explorer, and then click **Add Adapter Service Reference**. The Add Adapter Service Reference Plug-in opens.
2. From the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
3. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list, do one of the following:

Note

If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
Username	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive. Note If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.

4. Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).

Note

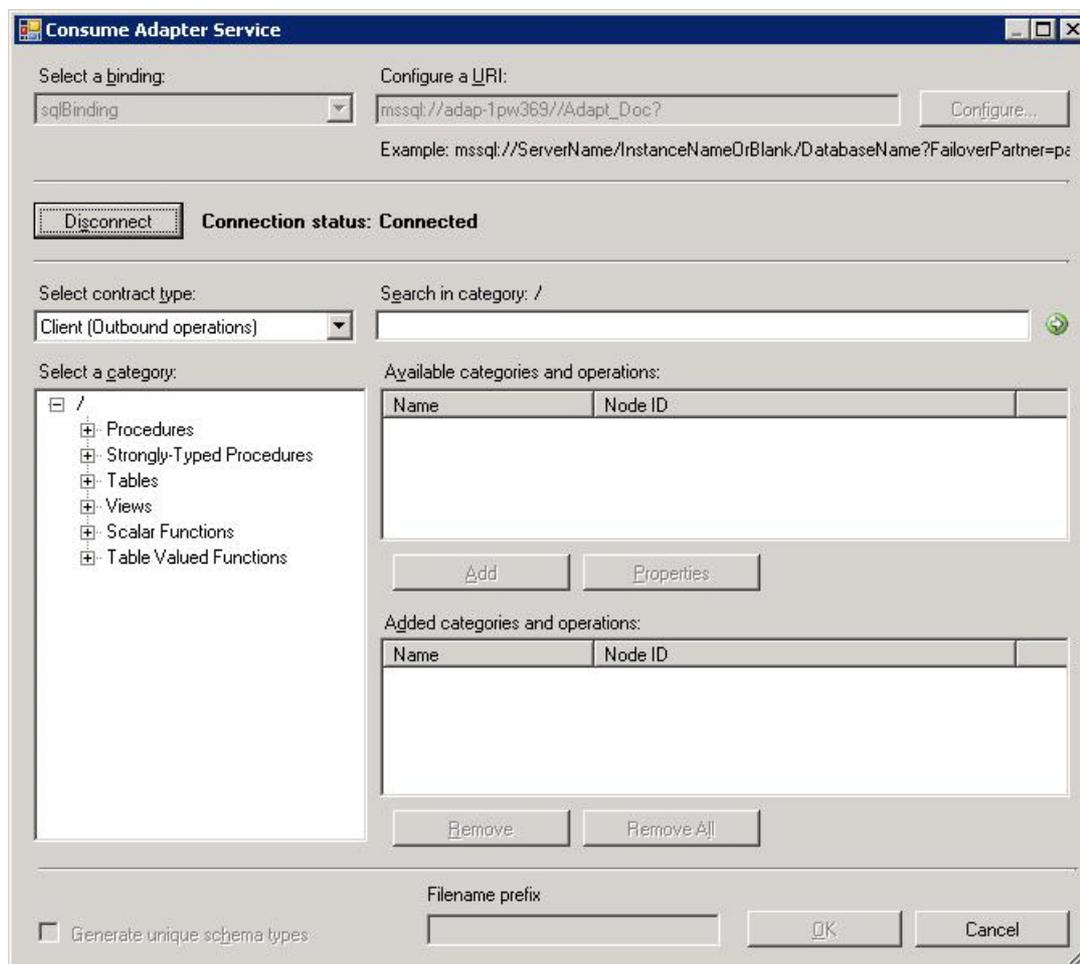
If the connection parameters contain any reserved characters, you must specify them as-is in the **URI Properties** tab, that is, without using any escape characters. However, if you specify the URI directly in the **Configure a URI** field and the connection parameters contain reserved characters, you must specify the connection parameters using proper escape characters.

Note

If you do not specify any values in the URI property tab, the Consume Adapter Service Add-in puts the URI as `mssql://.//`. In such a case, the adapter connects to the default database and the default database instance on the local computer.

5. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target.
6. Click **OK**.
7. Click **Connect**. After the connection is established, the connection status is shown as **Connected**.

The following figure shows the Consume Adapter Service Add-in immediately after the connection is established. The graphical user interface is same for the Add Adapter Service Reference Plug-in.



The Add Adapter Service Reference Plug-in displays different nodes containing various operations that can be performed on SQL Server. For example, the **Procedures** node contains all the procedures available for the database you connected to. Similarly, the **Tables** node contains all the tables in the database you connected to, and the operations that can be performed on a table. For more information about these nodes, see [Metadata Node IDs](#).

See Also

Other Resources

[Connecting to SQL Server in Visual Studio](#)

Browsing, Searching, and Retrieving Metadata for SQL Server Operations

This section provides information about how to use the Consume Adapter Service BizTalk Project Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Visual Studio Plug-in. By using these Visual Studio components, you can:

- Browse for operations for which to retrieve metadata.
- Search for operations for which to retrieve metadata.
- Add message schemas for selected operations and port binding configuration files to a BizTalk Server project when using the Consume Adapter Service Add-in.
- Add a WCF client class or a WCF service contract (interface) for selected operations and a configuration file (app.config) to a non-BizTalk programming project when using the Add Adapter Service Reference Plug-in.

Note

The Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, and the Add Adapter Service Reference Plug-in present essentially the same interface when you browse and search for operations, so all three components are covered in the same topics.

Prerequisites

You must connect to SQL Server before you can browse, search, or retrieve metadata for target operations. For information about how to connect to SQL Server when you use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in, see [Connecting to SQL Server in Visual Studio](#).

Browsing for Operations

You can use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in to browse for outbound and inbound operations that can be performed on SQL Server using the SQL adapter.

Outbound Operations

The SQL adapter enables clients to perform the following outbound operations.

- Insert, Select, Update, and Delete operations on tables and views.
- Set<column_name> operations on tables and views. This operation is exposed on tables that have columns of varchar(max), nvarchar(max), or varbinary(max). Such operations enable streaming of large objects.
- Stored procedures, both weakly and strongly-typed as operations.
- Scalar and table valued functions as operations.

The adapter also exposes the generic outbound operations such as **ExecuteReader**, **ExecuteScalar**, and **ExecuteNonQuery** at the root level.

Inbound Operations

The SQL adapter enables clients to perform the following inbound operations.

- **Polling** operation to receive polling-based data-change messages from SQL Server. The messages received for this operation are not strongly typed.
- **TypedPolling** operation to receive polling-based data-change messages from SQL Server. The messages received for this operation are strongly typed.
- **Notification** operation to receive query notifications from SQL Server.

Note

The adapter also supports an **XmlPolling** inbound operation to enable polling on SQL Server database using SELECT statements and stored procedures that contain a FOR XML clause. However, the adapter does not expose a specific inbound operation for this. For more information about XmlPolling, see [Receiving Polling Messages Using SELECT Statements with FOR XML Clause](#).

For more information about these operations, see [What Operations Can Be Performed Using the Adapter?](#).

Note

By using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, you can browse category and operation nodes using a Windows interface.

For more information about browsing metadata, see [How Does the Adapter Surface SQL Server Metadata?](#).

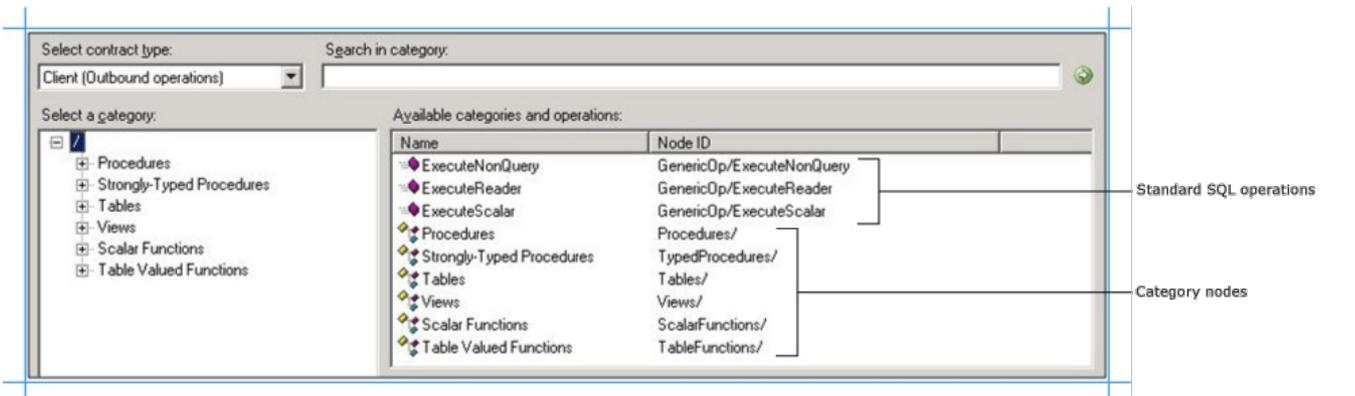
To browse outbound operations on SQL Server

1. Connect to SQL Server using the Consume Adapter Service Add-in, Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in. See [Connecting to SQL Server in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for outbound operations select **Client (Outbound operations)**.
3. The **Select a category** box lists the artifacts available in the SQL Server database you connect to. Click an artifact to view the operations available for that artifact in the **Available categories and operations** box.

Tip

You can directly go to the “immediate” category node or subcategory nodes in the tree, by typing the name of the artifact, while the focus is on the tree view in the **Select a category** box. For example, to jump to the **Employee** table node, keep the focus on the **Tables** node, and then type **Employee**.

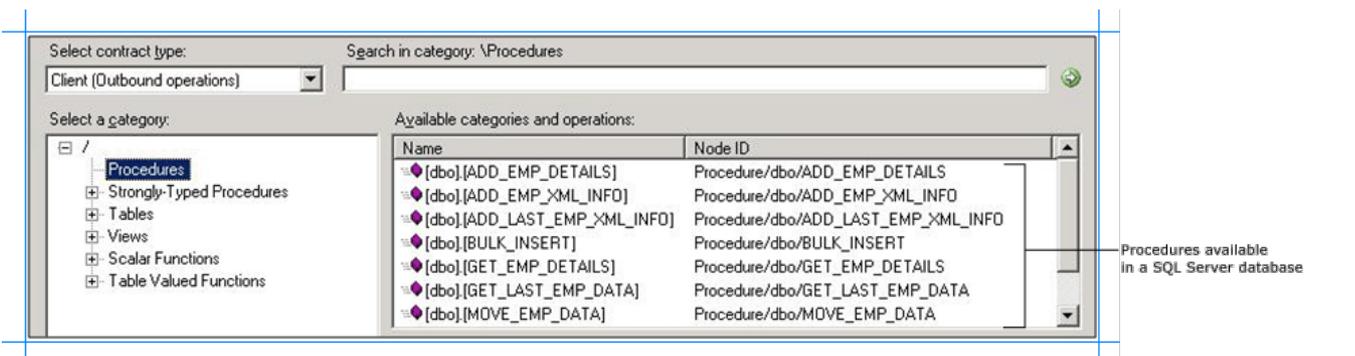
The following figure shows the Consume Adapter Service Add-in. The root node (/) is selected, and the general category nodes available under the root node are listed in the **Available categories and operations** box.



Note

The standard SQL Server operations such as ExecuteReader, ExecuteScalar, and ExecuteNonQuery are also available at the root level. For more information about these operations, see [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

4. To see the procedures available in the SQL Server database, click the **Procedures** node. In the following figure, the **Procedures** node is selected in the **Select a category** box, and the corresponding procedures are listed in the **Available categories and operations** box.

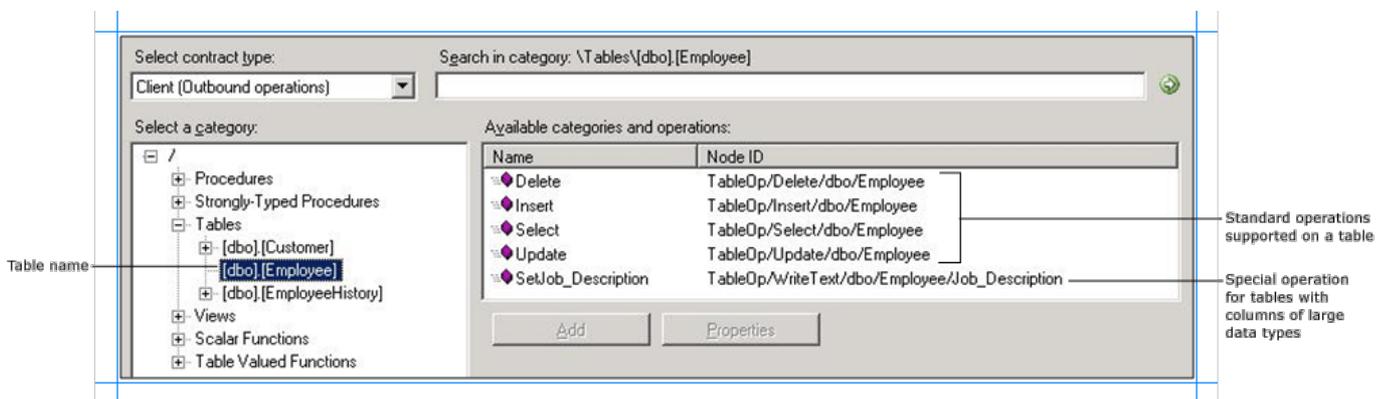


Note

The same set of procedures that are listed under the **Procedures** node are also available under the **Strongly-Typed Procedures** node. The difference is in the way schema is generated. For a procedure under the **Procedure** node, the schema is weakly typed. However, for a procedure under the **Strongly-Typed Procedure** node, the schema is strongly typed. Strongly-typed schema is useful if you want to map schema of one operation to another operation using BizTalk mapper because the schema is available to you at design-time while creating the BizTalk project. For weakly-typed procedures, the schema for the procedure is received at run time as part of the response message.

5. To see the tables in the SQL Server database, click the **Tables** node. Alternatively, expand the **Tables** node.
6. To see the operations supported on the table, click a table name.

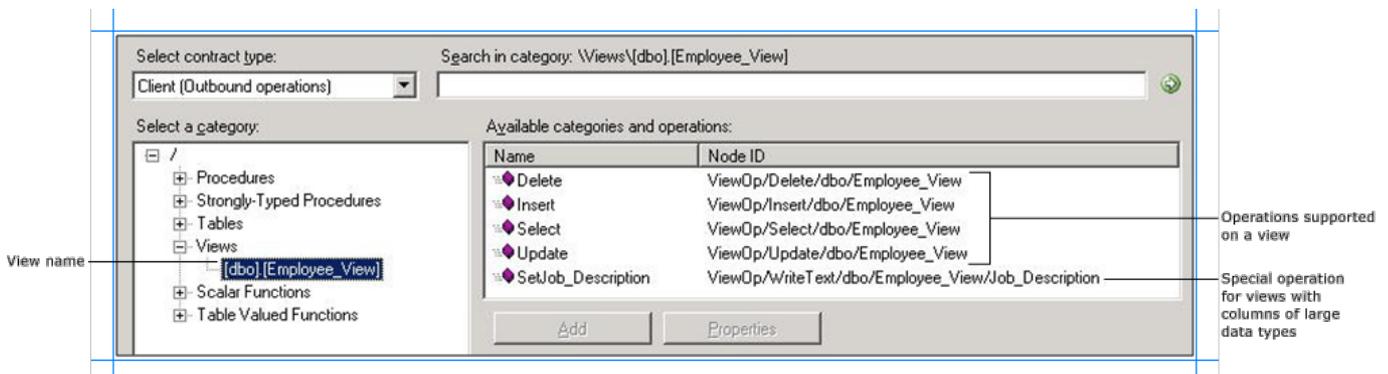
The following figure shows a list of tables in the **Select a category** box. The **Available categories and operations** box lists the operations supported for a selected table.



Note
 If a SQL Server table contains columns of type varchar(max), nvarchar(max), and varbinary(max), the adapter also exposes a specific operation to update data in that column. The name of this operation is Set<column_name>. For example, if the table has a column "Job_Description" of type varchar(MAX), the name of the operation is "SetJob_Description".

- To see the views in the SQL Server database, click the **Views** node. Alternatively, expand the **Views** node.
- To see the operations supported on the view, click a view name.

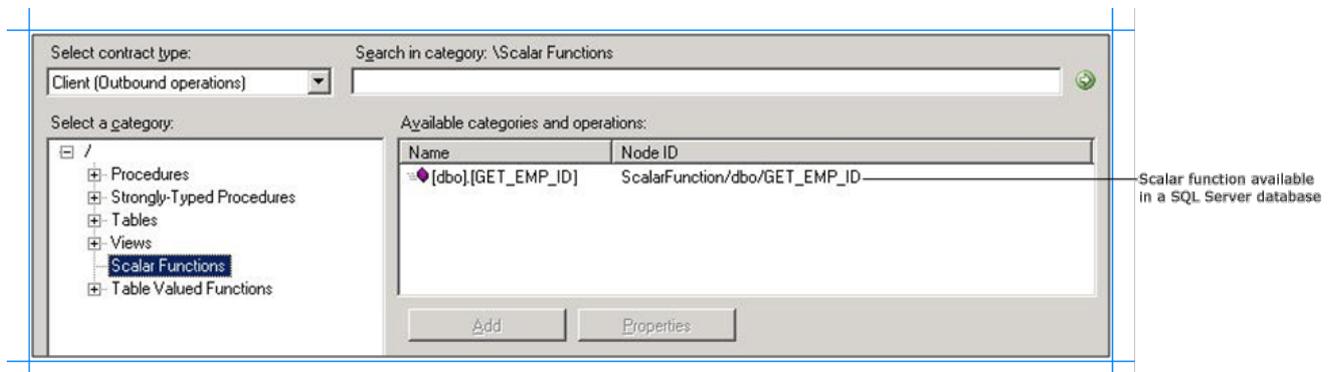
The following figure shows a list of views in the **Select a category** box. The **Available categories and operations** box lists the operations supported for a selected view.



Note
 If a view contains columns of type varchar(max), nvarchar(max), and varbinary(max), the adapter also exposes a specific operation to update data in that column. The name of this operation is Set<column_name>. For example, if the table has a column "Job_Description" of type varchar(max), the name of the operation is "SetJob_Description".

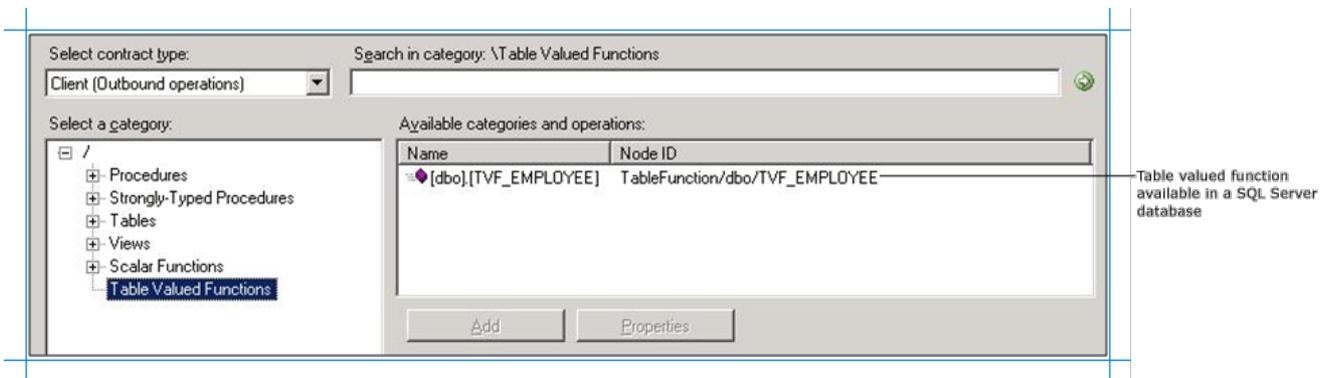
- To see the list of scalar functions defined in the SQL Server database in the **Available categories and operations** box, click the **Scalar Functions** node.

In the following figure, the **Scalar Functions** node is selected in the **Select a category** box, and the corresponding functions are listed in the **Available categories and operations** box.



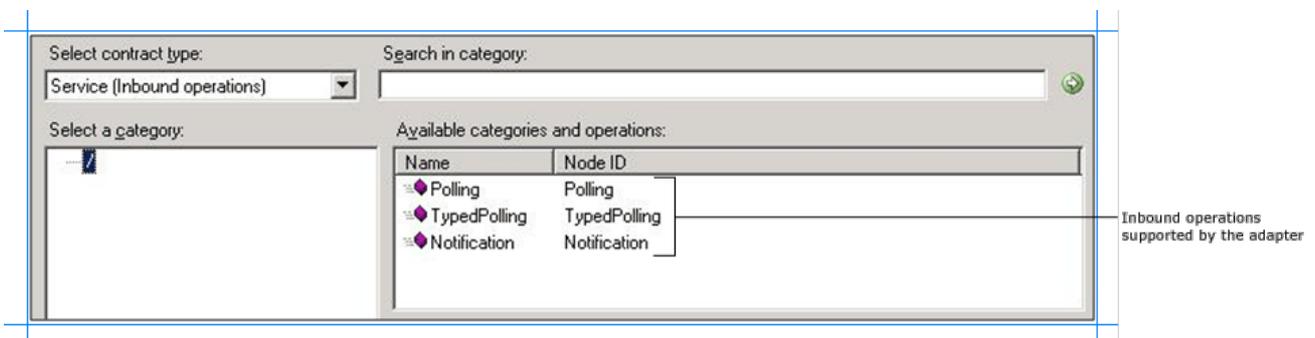
- To see the list of table valued functions defined in the SQL Server database in the **Available categories and operations** box, click the **Table Valued Functions** node.

In the following figure, the **Table Valued Functions** node is selected in the **Select a category** box, and the corresponding functions are listed in the **Available categories and operations** box.



To browse inbound operations on SQL Server

1. Connect to SQL Server using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in. See [Connecting to SQL Server in Visual Studio](#) for instructions.
2. From the **Select contract type** list, for inbound operations, select **Service (Inbound operations)**.
3. All the inbound operations supported by the SQL adapter are available at the root node. Click the root node (/) to view the available inbound operations.



Searching for Operations

You can use the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in to search for specific artifacts in the SQL Server database. When searching SQL Server metadata, the SQL adapter:

- Supports wildcard and escape characters in the search expression.
- Enables search immediately under the node at which the search operation is performed. For example, to search for a table, you must be searching under \Table. Multi-level search is not supported.

The following table lists the special characters that can be used for searching artifacts and their interpretation by the SQL adapter.

Special character	Interpretation	Example
_ (underscore)	Matches exactly one character	"A_" matches AB, AC, AD.
%	Matches zero or more characters	"A%" matches A, AB, AC.
[]	<ul style="list-style-type: none"> • Escapes the special meaning of % and _ • Specifies a range or set of characters to be present 	<ul style="list-style-type: none"> • [%]% matches all names that include a % symbol. • [a-f] matches all names that have characters between (and including) 'a' and 'f'. • [abc] matches all names that have characters 'a', 'b', and 'c'.
[^]	Specifies a range or set of characters not to be present	<ul style="list-style-type: none"> • [^a-f] matches all names that do not have characters between (and including) 'a' and 'f'. • [^abc] matches all names that do not have characters 'a', 'b', and 'c'.

Note

Escape character is a character that is put in front of a wildcard character to indicate that the wildcard should be interpreted as a regular character and not as a wildcard.

For more information, see [How Does the Adapter Surface SQL Server Metadata?](#).

To search metadata in SQL Server using the Consume Adapter Service Add-in, perform the following steps.

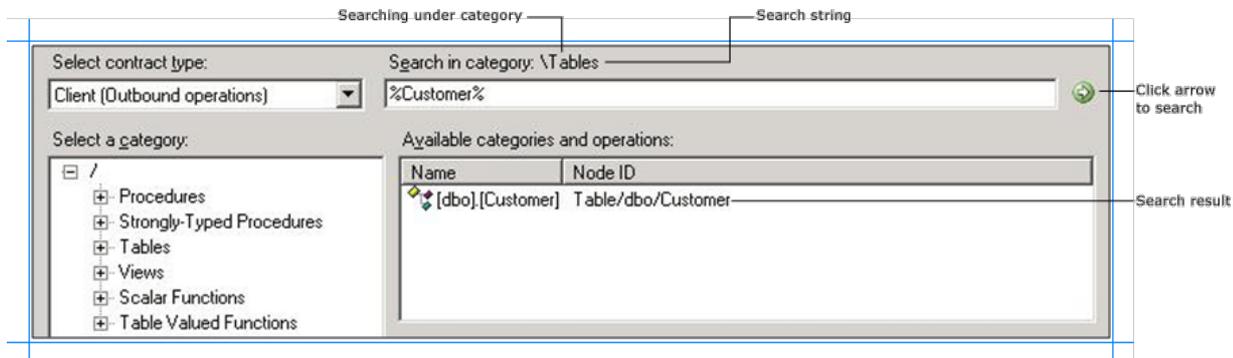
To search metadata in SQL Server

1. Connect to SQL Server using the Consume Adapter Service Add-in, the Add Adapter Metadata Wizard, or the Add Adapter Service Reference Plug-in. See [Connecting to SQL Server in Visual Studio](#) for instructions.
2. From the **Select contract type** list, select the type of contract based on whether you are searching for inbound or outbound operations.
3. In the **Select a category** box, click the category node under which you want to search for a specific artifact. For example, to search for a table, click the **Tables** node.
4. In the **Search in category** box, type a search expression to search for a specific artifact. For example, to search for tables that have "Customer" in their name, type **%Customer%**.

Note
The search string is case-sensitive.

5. To start the search, click the button with the right-arrow icon. After the search is complete, the **Available categories and operations** box lists the artifacts that satisfy the search criteria.

The following figure shows the SQL Server tables that contain "Customer" in their name.



Generating Schema for BizTalk Projects

You can use the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard to generate schema for selected SQL Server artifacts. Once you have browsed and searched for the artifacts you want to invoke, you can generate schema for those artifacts and send messages, conforming to the schema, to SQL Server.

Note
You can select category nodes to return all the operations in that category's sub-tree—for example, you can select an entire table (to generate schema for all the operations in the table) or select specific operations on a table (for example, Insert and Delete) to generate schema for only those operations on a table. For more information about the nodes, see [Metadata Node IDs](#).

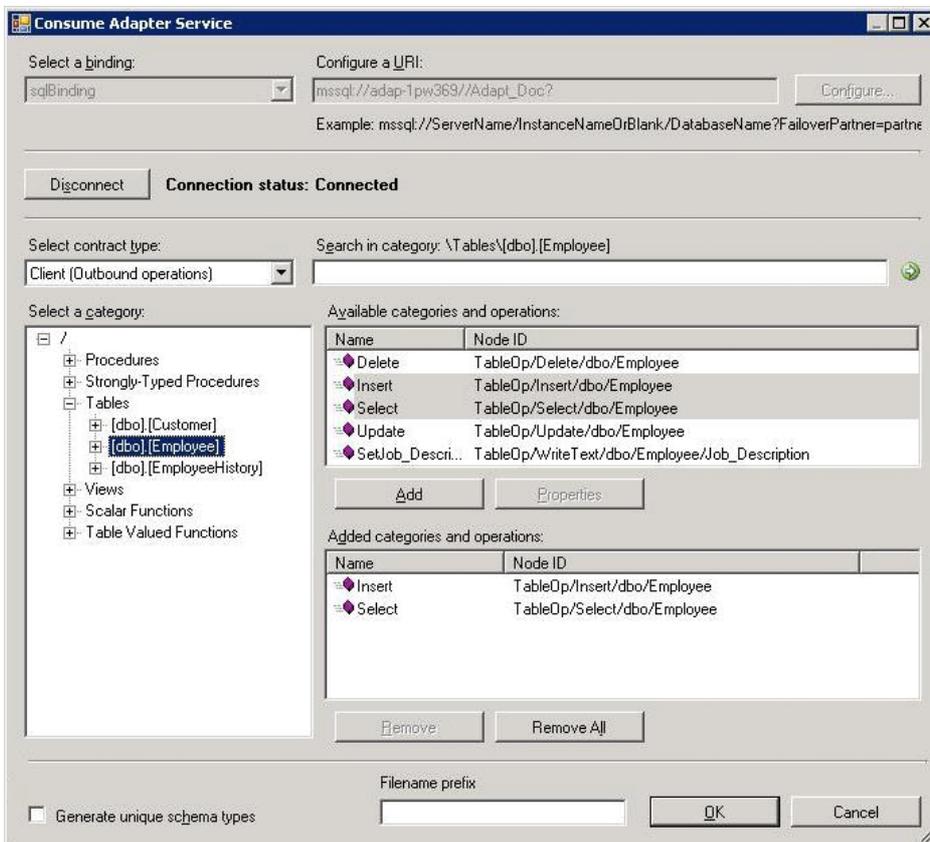
To generate schema for SQL Server artifacts

1. Connect to SQL Server using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard. See [Connecting to SQL Server in Visual Studio](#) for instructions.

Important
To generate schema for performing operations using BizTalk Server you must set the **EnableBizTalkCompatibilityMode** binding property to **True**. You must set this binding property while establishing a connection to the SQL Server database.

2. From the **Select contract type** list, select the type of contract based on whether you are generating schema for inbound or outbound operations.
3. Click the category node for which you want to generate metadata. For example, if you want to generate metadata for a table, click **Tables**.
4. Expand the category node, and select the specific item within that node for which you want to generate metadata. For example, to generate metadata for operations on the "CustomerTable" table, expand the **Tables** node, and then click **CustomerTable**.
5. In the **Available categories and operations** box, select the operations that you want to perform on SQL Server, and then click **Add**. The selected operations are listed in the **Added categories and operations** box. For example, to perform Insert and Select operations on the "CustomerTable" table, click the operation names, and then click **Add**.

The following figure shows the Consume Adapter Service Add-in, which lists the selected operations.



If you want to generate schema for multiple operations, there may be some duplicate element definitions among these schema that may cause failure in compiling the BizTalk project. For example, consider a scenario where you generate schema for an operation "Op1". The schema for "Op1" contains a parameter of data type "CT1". After generating the schema for "Op1" you close the Consume Adapter Service Add-in and re-open it to generate schema for another operation "Op2". Assume that "Op2" also contains a parameter of data type "CT1". After you exit the Consume Adapter Service Add-in and compile the project, you will get compilation errors because the complex data type "CT1" is defined twice in different XSD files. In such situations, we recommend the following:

- Generate schema for all the operations in a single run of Consume Adapter Service Add-in. This ensures that the Consume Adapter Service Add-in generates only one definition for the complex data type "CT1".
- If you want to generate schema for multiple operations across different runs of Consume Adapter Service Add-in, make sure you select the **Generate unique schema types** check box, so that the generated XSD files contain unique namespaces for the complex data type "CT1".

6. Click **OK**. The schema file is saved with an .xsd extension at the same location as the BizTalk project.

Note
If you are using the Consume Adapter Service Add-in to generate metadata for SQL Server artifacts, by default the files are created with a specific naming convention. The generated WSDL contains a **fileNameHint** annotation tag that includes the name that should be assigned to the XSD file. For example, a file name hint for a schema file for table operation follows the convention TableOperation.<schema>.<tablename>. If you want to customize the name of the generated XSD file, you can provide a prefix in the **Filename prefix** box. Finally, the name of an XSD file is arrived upon as Filename prefix + fileNameHint + unique integer (if required, to ensure that the filename is unique).

Note
The Consume Adapter Service Add-in also creates a binding file (an XML file) containing the binding properties that you specified when generating the schema for an operation and the SOAP action to invoke the operation. You can import this binding file in the BizTalk Server Administration console to create a WCF-Custom port or a BizTalk SQL adapter port with the connection URI, binding properties, and the SOAP action set. For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

You have successfully generated metadata for SQL Server artifacts. You can use the metadata to send messages to SQL Server to perform specific operations. See [Developing BizTalk Applications](#) for more information about how to perform these operations.

Generating a WCF Client or WCF Service Contract Using the Add Adapter Service Reference Plug-in

You can use the Add Adapter Service Reference Plug-in to generate either WCF client code for outbound operations or WCF service code for the inbound operations.

To generate WCF client class or service contract for SQL Server operations

1. In the Add Adapter Service Reference Plug-in, from the **Select contract type** drop-down list, select the type of contract based on whether you will be performing inbound or outbound operations.

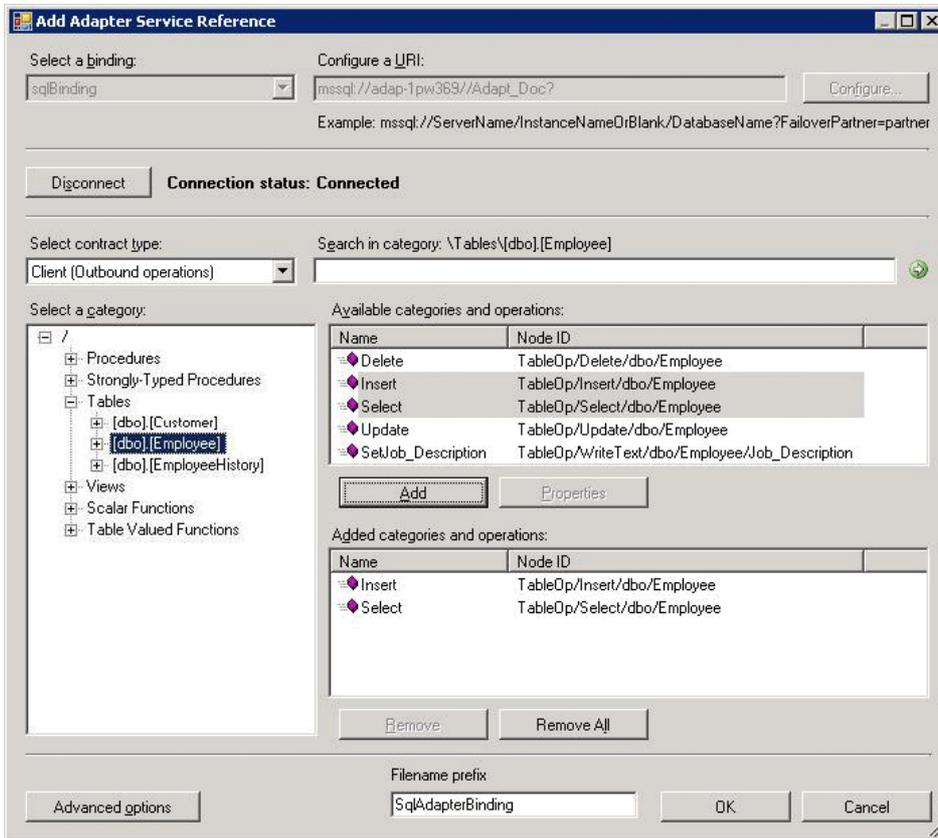
2. Browse or search for categories (such as a database table) or for the specific operations for which you want to generate a WCF client (or WCF service contract).

For example, to browse for the operations in the Employee table, in the **Select a category** box:

- a. Expand the root node (/) to see the categories under which the operations are surfaced for a SQL Server database.
- b. Under the root node, expand the **Tables** node to see the available tables.

3. Click the **Employee** table node, and in the **Available categories and operations** box, select the operations or categories for which you want to generate a WCF client (or WCF service contract), and then click **Add**. The selected operations are listed in the **Added categories and operations** box.

The following figure shows the Add Adapter Service Reference Plug-in with the Insert and Select operations for the Employee table selected.



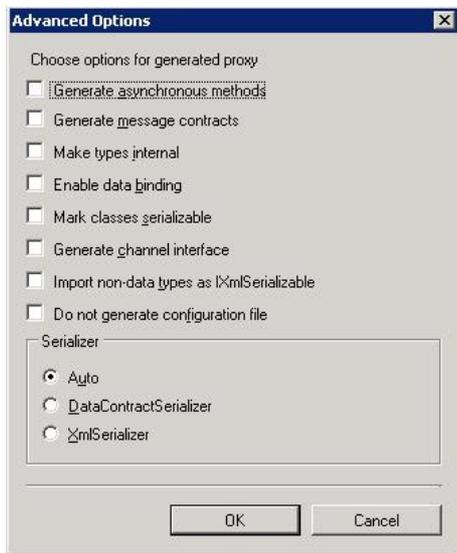
Important

Depending on the outbound operations (or categories) that you select, more than one WCF client class may be generated. For more details, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

4. For most scenarios the default serialization options are sufficient; however, if needed, you can control several aspects about the code that is generated and the type of serializer that is used. To set these options:

- a. Click **Advanced Options** to open the **Advanced Options** box.
- b. In the **Advanced Options** box under **Choose options for generated proxy**, select the options that you want. For example, you can select whether asynchronous methods are generated for the WCF client or disable the generation of a configuration file.
- c. Under **Serializer** select the serializer that should be used.

The following figure shows the **Advanced Options** box with the default selections (**Auto** is selected for the serializer and no other options are selected).



The options that you can configure in the **Advanced Options** box are equivalent to some of the options available when you use the ServiceModel Metadata Utility Tool (svcutil.exe). For more information about these options, see "ServiceModel Metadata Utility Tool (Svcutil.exe)" at <http://go.microsoft.com/fwlink/?LinkId=99821>.

5. Click **OK**. The Add Adapter Service Reference Plug-in saves the WCF client class (or WCF service interface) and helper code for the operations and categories that you have selected in your project directory. By default, a configuration file is also saved. Slightly different files are generated for inbound and outbound operations; for more information, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

You can select any node that is listed in the **Available categories and operations** box. If you select a category node then all of the operations available under that node and its sub-nodes will be selected. For example, to generate a WCF client for all of the operations surfaced for the Employee table, you can select the Employee node.

See Also

Other Resources

[Retrieving Metadata for SQL Server Operations in Visual Studio](#)

Working with BizTalk Adapter for SQL Server Binding Properties

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for SQL Server surfaces several binding properties. By setting these properties, you can control some of the adapter's behavior. This section describes the binding properties exposed by the SQL adapter. It also shows how you can access them by using .NET programming or by setting properties on a BizTalk Server physical port binding.

The Adapter Binding Properties

The following table shows SQL adapter binding properties grouped by category. The category refers to the node under which each binding property appears in the dialog boxes that are presented by different applications to configure the adapter (or binding).

Binding Property	Category	Description	.NET Type
XmlStoredProcedureRootName	(FOR XML)	Specifies the name of the root node for the response schema for stored procedures that have a FOR XML clause in the SELECT statement. This root node encapsulates the XML response received from SQL Server after executing such stored procedures. You must add this root node to the response schema as described in the topic Executing Stored Procedures Having a FOR XML Clause . ◆ Important You must set this binding property while executing stored procedures with FOR XML clause.	string
XmlStoredProcedureRootNamespace	(FOR XML)	Specifies the target namespace of the root node for the response schema for stored procedures that have a FOR XML clause in the SELECT statement.	string
CloseTimeout	(General)	The WCF connection close timeout. The default is 1 minute.	System.TimeSpan
Name	(General)	A read-only value that returns the name of the file generated by the Add Adapter Service Reference Visual Studio Plug-in to hold the WCF client class. The Add Adapter Service Reference Plug-in forms the file name by appending "Client" to the value of the Name property. The default value for this property is "SqlAdapterBinding"; for this value, the generated file will be named "SqlAdapterBindingClient".	string
OpenTimeout	(General)	Specifies the WCF connection open timeout. The default is 1 minute. ◆ Important The SQL adapter always uses OpenTimeout to set the connection open timeout when it opens a connection to SQL Server. The adapter ignores any timeout (System.TimeSpan) parameters passed when you open a communication object. For example, the adapter ignores any timeout parameters passed when opening a channel.	System.TimeSpan

ReceiveTimeout	(General)	<p>Specifies the WCF message receive timeout. Essentially, this means the maximum amount of time the adapter waits for an inbound message. The default is 10 minutes.</p> <p>◆Important</p> <p>For inbound operations such as polling, we recommend setting the timeout to the maximum possible value, which is 24.20:31:23.6470000 (24 days). When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.</p>	System.TimeoutSpan
SendTimeout	(General)	<p>Specifies the WCF message send timeout. The default is 1 minute.</p>	System.TimeoutSpan
EnableBizTalkCompatibilityMode	BizTalk	<p>Indicates whether the adapter is being used with BizTalk Server or a .NET application.</p> <ul style="list-style-type: none"> When using the adapters from BizTalk Server (or generating metadata for operations on SQL Server using the adapter in a BizTalk project), you must always set the property to True. This ensures that the schema generated for System.Data.DataSet is in a format compatible with BizTalk Server, otherwise your BizTalk project will fail to compile. When using the adapters from Visual Studio in a .NET application, you must set the property to False if you want to use the response as a DataSet. This ensures that the schema generated for System.Data.DataSet is in a format compatible with the WCF DataContractSerializer. 	boolean (System.Boolean)
BatchSize	Buffering	<p>Specifies the batch size for multiple record Insert, Update, and Delete operations on a table or view in a SQL Server database. The default is 20. For values of BatchSize greater than one, the SQL adapter batches the specified number of records into a single call. A higher value may improve performance, but affects memory consumption.</p>	int (System.Int32)
ChunkSize	Buffering	<p>Specifies the buffer size used for Set<column_name> operations. The default is 4194304 bytes. A higher value may improve performance, but affects memory consumption.</p> <p>📌Note</p> <p>For more information about Set<column_name> operations, see Operations on Tables and Views That Contain Large Data Types.</p>	int (System.Int32)
Encrypt	Connection	<p>Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client. Default is false.</p>	boolean (System.Boolean)

MaxConnectionPoolSize	Configuration property	<p>Specifies the maximum number of connections allowed in a connection pool for a specific connection string. The default is 100. This property is used for performance tuning.</p> <p>◆ Important You must set MaxConnectionPoolSize judiciously. It is possible to exhaust the number of connections available, if this value is set too large.</p>	int (System.Int32)
WorkstationId	Configuration property	<p>Specifies a unique ID for the workstation (client computer) that connects to the SQL Server database using the SQL adapter. The value for this binding property, if specified, is used for the Workstation ID keyword of SqlConnection.ConnectionString property. For more information, see "SqlConnection.ConnectionString Property" at http://go.microsoft.com/fwlink/?LinkID=196958.</p>	string
EnablePerformanceCounters	Configuration property	<p>Specifies whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter. The default is False; performance counters are disabled. The LOB latency performance counter measures the total time spent by the SQL adapter in making calls to the SQL Server database.</p> <p>For more information about performance counters for the SQL adapter, see Using Performance Counters.</p>	int (System.Int32)
InboundOperationType	Configuration property	<p>Specifies whether you want to perform Polling, TypedPolling, XmlPolling, or Notification inbound operation. Default is Polling.</p> <p>For more information about Polling, TypedPolling, and XmlPolling see Support for Polling. For more information about Notification, see Receiving Query Notifications.</p>	enum
UseDatabaseNameInXsdNamespace	Configuration property	<p>Specifies whether the XSD generated for a particular artifact contains the database name. Set this to True to include the database name. Otherwise, set this to False. Default is False.</p> <p>This is useful in scenarios where a single application wants to execute operations on identically named artifacts with different metadata in different databases. If there is no database name in the namespace, the generated metadata will conflict. By setting this binding property you can include the database name in the namespace, thereby making them unique. Here is an example highlighting the change in namespace.</p> <p>UseDatabaseNameInXsdNamespace = False</p> <div data-bbox="217 1368 1453 1449" style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee</pre> </div> <p>UseDatabaseNameInXsdNamespace = True</p> <div data-bbox="217 1520 1453 1601" style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>http://schemas.microsoft.com/Sql/2008/05/TableOp/MyDatabase/dbo/Employee</pre> </div> <p>Notice that the database name is included in the namespace when the binding property is set to True.</p>	enum

Allow IdentityInsert	Microsoft SQL Server	<p>Specifies whether the adapter can insert values for identity columns during Insert and Update operations. Set this property to True to insert or update values for identity columns. Otherwise set this to False. Default is False.</p> <p> Note Setting this property to True translates to the adapter using "SET IDENTITY_INSERT <table_name> ON". For more information, see http://go.microsoft.com/fwlink/?LinkId=124620.</p> <p>While using this binding property, you must consider the following points:</p> <ul style="list-style-type: none"> • The adapter does not validate the value you are passing for the identity column. For example, if a table has an identity column that has "Identity Seed" set to 100 and "Identity Increment" set to 1, and the adapter client passes a value, say 95, for the identity column, the adapter simply passes on this value to SQL Server. • Even if you set AllowIdentityInsert to True, it is not mandatory for an adapter client to specify a value for the identity column in the request message. If a value is present for the identity column, the adapter passes it on to SQL Server. If a value is not present, SQL Server will insert a value based on the specification for the identity column. 	Boolean
NotificationStatement	Microsoft SQL Server	<p>Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications. Note that you must specifically specify the column names in the statement as shown in this SELECT statement.</p> <pre>SELECT Employee_ID,Designation FROM dbo.Employee WHERE Status=0</pre> <p> Note You must specify the database object name along with the schema name. For example, <code>dbo.Employee</code>.</p> <p>The adapter gets a notification message from SQL Server only when the result set for the specified SQL statement changes.</p>	String
NotifyOnListenerStart	Microsoft SQL Server	<p>Specifies whether the adapter sends a notification message to the adapter clients, informing that the receive location is running, when the listener starts. Default is True.</p> <p>The notification message you receive looks like the following:</p> <pre><?xml version="1.0" encoding="utf-8" ?> <Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/"> <Info>ListenerStarted</Info> <Source>SqlBinding</Source> <Type>Startup</Type> </Notification></pre>	Boolean

PolledDataAvailableStatement	PolledDataAvailableStatement	<p>Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database. The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the SQL statement specified for the PolledDataAvailableStatement binding property. If the first cell of the result contains a positive value, the adapter executed the polling statement. Following are examples of some valid statements that you can specify for this binding property:</p> <ul style="list-style-type: none"> If you are specifying a SELECT statement: <pre style="border: 1px solid black; padding: 5px; margin: 10px 0;">SELECT COUNT(*) from <table_name></pre> <ul style="list-style-type: none"> If you are specifying a stored procedure, your stored procedure might be defined as: <pre style="border: 1px solid black; padding: 5px; margin: 10px 0;">CREATE PROCEDURE <procedure_name> AS BEGIN SELECT COUNT(*) FROM <table_name> END GO</pre> <p>or</p> <pre style="border: 1px solid black; padding: 5px; margin: 10px 0;">CREATE PROCEDURE <procedure_name> AS BEGIN DECLARE @count int SELECT @count = SELECT(*) FROM <table_name> SELECT @count END GO</pre> <p>If you are using a stored procedure, you would specify the PolledDataAvailableStatement as EXEC <procedure_name>.</p> <p>◆ Important The statement you specify for this binding property is not executed within <i>an adapter initiated</i> transaction, and might be called multiple times before the actual polling statement is executed (even if executing the statement indicated that there are rows available for polling).</p>	string
PollingIntervalSeconds	PollingIntervalSeconds	<p>Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter is inactive for the remaining time in the interval.</p>	int (System.Int32)

PollingStatement	PolledStatement	<p>Specifies the SQL statement to poll a SQL Server database table. You can specify a simple SELECT statement or stored procedures for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.</p> <p>You can specify any number of SQL statements separated by a semicolon. You can use the polling statement to read or update data in a SQL Server database table. The SQL adapter executes the polling statements inside one transaction. When the adapter is used with BizTalk Server, the same transaction is used to submit messages from SQL Server to the BizTalk message box.</p>	string
PollWhileDataFound	PollWhileDataFound	<p>Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false.</p> <p>Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for PolledDataAvailableStatement returns that data is available for polling. The adapter then executes the statement specified for the PollingStatement binding property. Assuming that the adapter takes just 10 seconds to execute the polling statement, it will now have to wait for 50 seconds before executing the PolledDataAvailableStatement again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the PollWhileDataFound binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.</p>	boolean (System.Boolean)
UseAmbientTransaction	UseAmbientTransaction	<p>Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller. Default is true, which means that the adapter always performs the operations in a transactional context. If there are other resources participating in the transaction, and SQL Server also joins the transaction, the transaction gets elevated to an MSDTC transaction.</p> <p>However, there can be scenarios where you do not want the adapter to perform operations in a transactional context. For example:</p> <ul style="list-style-type: none"> • While performing a simple Select operation on a SQL Server database • While specifying a polling statement that performs a Select operation, and does not involve any changes to the table either through a Delete statement or by invoking a stored procedure. <p>Both these operations do not make any updates to the database table and, hence, elevating these operations to use an MSDTC transaction can be a performance overhead. In such scenarios, you can set the binding property to false so that the SQL adapter does not perform the operations in a transactional context.</p> <p> Note Not performing operations in a transactional context is advisable only for operations that do not make changes to the database. For operations that update data in the database, we recommend setting the binding property to true; otherwise you might either experience message loss or duplicate messages, depending on whether you are performing inbound or outbound operations.</p>	boolean (System.Boolean)

How Do I Set SQL Server Binding Properties?

You can set the SQL Server binding properties when you specify a connection to SQL Server. For information about how to set binding properties when you:

- Use the Consume Adapter Service BizTalk Project Add-in or the Add Adapter Service Reference Visual Studio Plug-in, see [Connecting to SQL Server in Visual Studio](#).

Important

While using the Consume Adapter Service Add-in or the Add Adapter Service Reference Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file (an XML file) or the app.config file respectively. You must manually add the binding property and its value in the binding file or the app.config file, if required.

- Configure a send port or receive port (location) in a BizTalk Server solution, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).
- Use the WCF channel model in a programming solution, see [Creating a Channel](#).
- Use the WCF service model in a programming solution, see [Specifying a Client Binding for the SQL Adapter](#).

See Also

Other Resources

[Development](#)

Before Developing Applications Using the SQL Adapter

This section provides information about what you must do before developing applications using the SQL adapter.

In This Section

- [Configure MSDTC on SQL Server and Adapter Client](#)

Configure MSDTC on SQL Server and Adapter Client

The operations performed on SQL Server using the SQL adapter (through BizTalk Server, the WCF service model, or the WCF channel model) can be performed within a transaction scope. If the client program has more than one transactional resource as part of the same transaction, the transaction gets elevated to an MSDTC transaction. To enable the adapter to perform operations within the scope of an MSDTC transaction, you must configure MSDTC both on the computer running the SQL adapter and SQL Server. Also, you must add MSDTC to the exceptions list of Windows Firewall. This section provides information about how to perform these tasks on computers running the adapter client and SQL Server.

Note

Performing operations on SQL Server using BizTalk Server always involves two resources—the adapter connecting to SQL Server and the BizTalk Message Box residing on SQL Server. Hence, all operations performed using BizTalk Server are performed within the scope of an MSDTC transaction. So, to use the SQL adapter with BizTalk Server, you must always enable MSDTC.

Note

For operations where the adapter client does not write any data to the SQL Server database, such as a Select operation, you might not want the additional overhead of performing the operations inside a transaction. In such cases, you can configure the SQL adapter to perform operations without a transactional context by setting the **UseAmbientTransaction** binding property to **false**. For more information about the binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). In such cases, you do not need to configure MSDTC as well.

Configuring MSDTC

- To configure MSDTC on computers running Windows Vista or Windows Server 2008, refer to <http://go.microsoft.com/fwlink/?LinkId=139016>.

Enabling Firewall Exceptions for MSDTC

- To enable firewall exceptions for MSDTC on computers running Windows Vista or Windows Server 2008, refer to <http://go.microsoft.com/fwlink/?LinkId=138685>.

See Also

Other Resources

[Before Developing Applications Using the SQL Adapter](#)

Developing BizTalk Applications

Developing BizTalk applications involves creating a BizTalk project in Visual Studio and using the Consume Adapter Service BizTalk Project Add-in or Add Adapter Metadata Wizard to generate XML schema. Once you have generated the schema, you can either use Content-Based Routing (CBR) or create BizTalk orchestrations to send and receive messages that conform to the generated schema.

CBR can be used in scenarios where the messages being sent to SQL Server do not require any intensive processing. For example, if you know that the receive port will receive messages only of a certain type, you can add a filter to the send port to route messages that match the filter expression to the send port.

In BizTalk orchestrations, you create send and receive ports to send and receive messages to and from the WCF-Custom adapter, which in turn sends the messages to BizTalk Server. This section provides information about using BizTalk orchestrations to perform operations on SQL Server using the SQL adapter. The SQL adapter in turn uses the WCF-Custom adapter, which can interact with a WCF binding.

◆ Important

To use the SQL adapter with Microsoft BizTalk Server, you must always set the **EnableBizTalkCompatibilityMode** binding property to **True**. For information about how to set binding properties, see [Specifying Binding Properties](#).

In This Section

- [Before You Develop BizTalk Applications](#)
- [Building Blocks for Developing BizTalk Applications](#)
- [Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server](#)
- [Performing Operations on Tables and Views with Large Data Types by Using BizTalk Server](#)
- [Executing Stored Procedures in SQL Server by Using BizTalk Server](#)
- [Executing Stored Procedures With a Single XML Parameter](#)
- [Executing Stored Procedures Having a FOR XML Clause](#)
- [Performing Composite Operations on SQL Server by Using BizTalk Server](#)
- [Invoking Scalar Functions in SQL Server by Using BizTalk Server](#)
- [Invoking Table-Valued Functions in SQL Server by Using BizTalk Server](#)
- [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using BizTalk Server](#)
- [Polling SQL Server by Using the SQL Adapter with BizTalk Server](#)
- [Receiving Query Notifications by Using BizTalk Server](#)

See Also

Other Resources

[Development](#)

Before You Develop BizTalk Applications

This section provides information about what you must do before developing BizTalk applications using the SQL adapter. This section also points you to the BizTalk Server tools that are used to develop BizTalk applications.

In This Section

- [Create a Strong-Name Key File](#)
- [Learn About BizTalk Server Tools](#)

Create a Strong-Name Key File

You must create a strong-name key file to build projects in Microsoft Visual Studio. A strong name consists of the project's identity—its simple text name, version number, and culture information (if provided)—plus a public key and a digital signature. The strong-name key file contains the public key and the private key.

Important

Creating a strong-name key file is a one-time task. You can use the same key for all the BizTalk applications you develop.

Prerequisites

You must have Microsoft Visual Studio installed on the computer where you want to create a strong-name key.

To create a strong-name key file

1. Click **Start**, point to **Programs**, point to **Microsoft Visual Studio 2010**, point to **Visual Studio Tools**, and then click **Visual Studio Command Prompt (2010)**.
2. At the command prompt, navigate to the location where you want to create the key file. For example, type **cd C:\Sample**, and then press ENTER.
3. At the command prompt, type **sn -k <key file name>.snk**, and then press ENTER.

Note

You should receive a message at the command prompt stating that the key pair was written to the strong-name key file.

4. At the command prompt, type **exit**, and then press ENTER.

See Also

Concepts

[Learn About BizTalk Server Tools](#)

Other Resources

[Before You Develop BizTalk Applications](#)

Learn About BizTalk Server Tools

The topics on how to use the SQL adapter in [Developing BizTalk Applications](#) are written with the assumption that you have working knowledge of a number of BizTalk Server tools. You will use the following tools to develop BizTalk applications using the SQL adapter:

- Microsoft Visual Studio 2010
- Orchestration Designer
- Pipeline Designer
- BizTalk Mapper
- BizTalk Server Administration console

Prerequisites

You must install Microsoft BizTalk Server before you can access the BizTalk Server tools.

BizTalk Server Tools

The following table includes topics in the BizTalk Server documentation that explain how to use each of the listed tools.

Tool	Topics in BizTalk Server Documentation
Visual Studio	<ul style="list-style-type: none"> • Using Visual Studio (http://go.microsoft.com/fwlink/?LinkId=107180) • Working with BizTalk Projects (http://go.microsoft.com/fwlink/?LinkId=107181) • Deploying BizTalk Assemblies from Visual Studio into a BizTalk Application (http://go.microsoft.com/fwlink/?LinkId=107182) <p>Learn more about Visual Studio:</p> <ul style="list-style-type: none"> • Introduction to Solutions, Projects, and Items (http://go.microsoft.com/fwlink/?LinkId=196962) • Solution Explorer Concepts (http://go.microsoft.com/fwlink/?LinkId=196963)
Orchestration Designer	<ul style="list-style-type: none"> • Creating Orchestrations Using Orchestration Designer (http://go.microsoft.com/fwlink/?LinkId=107184)
Pipeline Designer	<ul style="list-style-type: none"> • Creating Pipelines Using Pipeline Designer (http://go.microsoft.com/fwlink/?LinkId=107185)
BizTalk Mapper	<ul style="list-style-type: none"> • Creating Maps Using BizTalk Mapper (http://go.microsoft.com/fwlink/?LinkId=107186)
BizTalk Server Administration console	<ul style="list-style-type: none"> • Using the BizTalk Server Administration Console (http://go.microsoft.com/fwlink/?LinkId=107187)

See Also

Tasks

[Create a Strong-Name Key File](#)

Other Resources

Building Blocks for Developing BizTalk Applications

To perform operations on SQL Server by using the SQL adapter with BizTalk Server, you must perform a set of design-time and run-time tasks using Visual Studio and BizTalk Server Administration console respectively. This section provides an overview of these tasks. All the topics in this section, which demonstrate how to perform specific operations on SQL Server using BizTalk Server, are modeled on these high-level tasks.

Using Visual Studio

1. **Create BizTalk project, and generate schema.** You must create a BizTalk project in Visual Studio, and generate schema for the operation that you will perform on SQL Server. For example, if you want to insert records in a SQL Server table, you must generate schema for the Insert operation for that table. To generate schema, you must use the Consume Adapter Service Add-in or Add Adapter Metadata Wizard. For more information, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#).
2. **Set up an orchestration.** Once you have generated the schema, you must set up an orchestration by using the Orchestration Designer. For a basic orchestration, you add the Send and Receive shapes along with the Send and Receive logical ports. In later steps, you map these logical ports to physical ports by using the BizTalk Server Administration console. The orchestration uses these ports to pick messages that an adapter client sends. The orchestration then passes the messages to SQL Server. Once SQL Server sends a response, the orchestration passes the response back to the adapter client.
3. **Create messages, and link to schema.** In your orchestration, you must create messages that will be mapped to the schema you generated in the first step. Typically, you create a request message and a response message. These messages are mapped to the corresponding request and response schemas.
4. **Map message shapes to messages and ports.** In your orchestration, you must now map each shape that you added in the second step to messages that you created in the third step. You must also map a message shape to the port on which that message will be sent.

For example, if the first shape in your orchestration is a Receive shape that will receive a message, you map this shape to a request message and the port that sends the request message.
5. **Build and deploy the BizTalk project.** After you have set up the orchestration and mapped messages, ports, and schemas, you must build the BizTalk solution. For building a project in Visual Studio, you need an assembly key file. After you successfully build the solution, you must deploy the solution.

Note

More detailed description of these high-level tasks, including procedural information, is provided in various topics of this section.

Once you have successfully built and deployed the BizTalk project, your tasks in Visual Studio are accomplished. You must now perform certain tasks using the BizTalk Server Administration console.

Using the BizTalk Server Administration Console

1. **Configure the application.** The BizTalk project you deployed by using Visual Studio shows up in the BizTalk Server Administration console as an orchestration. You must configure this orchestration by mapping the logical ports you created in Visual Studio to physical ports that you must now create using the BizTalk Server Administration console.

On the physical ports, you must specify an "action" or "action mapping". This action corresponds to the operation you want to perform on SQL Server. You need to specify the action if you are not using dynamic actions. For more information about actions, see [Specifying SOAP Action](#).

2. **Start the application.** After the application is configured, you must start the application, and drop request messages at a defined file location. The orchestration consumes the request messages, passes them to SQL Server, and receives a

response. This response is available to the adapter client at another defined file location.

To accomplish these high-level tasks, you must also perform other tasks. For example, when you use the Consume Adapter Service Add-in or Add Adapter Metadata Wizard to generate the schema, you must specify a connection URI to connect to SQL Server. This section provides information on such repetitive tasks that you must perform as you develop BizTalk applications using the SQL adapter.

In This Section

- [Adding the SQL Adapter to BizTalk Server Administration Console](#)
- [Specifying the Connection URI](#)
- [Specifying Credentials for SQL Server](#)
- [Specifying Binding Properties](#)
- [Specifying SOAP Action](#)
- [Manually Configuring a Physical Port Binding to the SQL Adapter](#)
- [Configuring a Physical Port Binding Using a Port Binding File](#)
- [Configuring Dynamic Ports](#)
- [Reusing Adapter Bindings](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Adding the SQL Adapter to BizTalk Server Administration Console

The SQL adapter can be used in BizTalk either as a WCF-Custom port or a WCF-SQL port. If you want to use the SQL adapter through a WCF-Custom port, you do not need to add the WCF-Custom port to the BizTalk Server Administration console because the WCF-Custom port is added to the BizTalk Server Administration console by default. However, if you want to use the SQL adapter through a WCF-SQL port, you must first add the WCF-SQL adapter to the BizTalk Server Administration console.

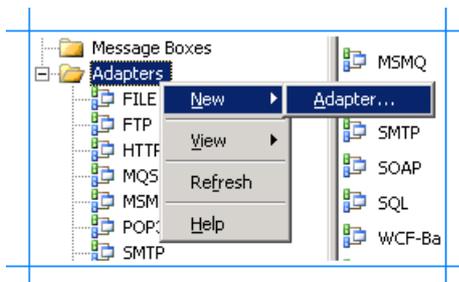
This topic provides instructions on how to add the WCF-SQL adapter to the BizTalk Server Administration console.

◆ Important

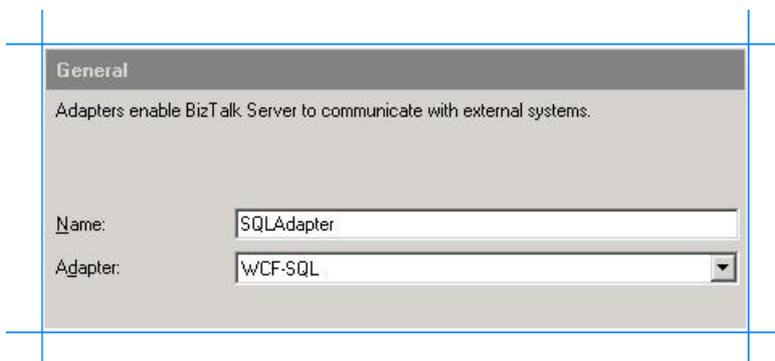
You need not perform these tasks if you want to configure a WCF-Custom port for the SQL adapter.

To add the SQL Adapter

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, expand **Platform Settings**, and then click **Adapters**.
3. Right-click **Adapters**, point to **New**, and click **Adapter**.



4. In the **Adapter Properties** dialog box, specify a name for the adapter and from the **Adapter** list, select **WCF-SQL**.



5. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying the Connection URI

A connection URI is a connection string that contains parameters required to connect to SQL Server. While using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard in Visual Studio, you must specify the URI to connect to SQL Server to generate the metadata. While configuring a send or receive port using the BizTalk Server Administration console, you must specify the URI to connect to SQL Server to perform operations.

Specifying the Connection URI from Visual Studio

From Visual Studio, you can specify the connection URI using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify the connection URI using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **sqlBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab. From the **Client credential type** list, do one of the following:

 Note
If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication .

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
User name	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive.  Note If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.

6. Click the **URI Properties** tab, and specify values for different parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
7. Click the **Binding Properties** tab, and specify values for the binding properties, if any, which are required before generating the schema. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
8. Click **OK**.

To specify the connection URI using Add Adapter Metadata Wizard

1. Right-click the BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Wizard, select **WCF-SQL**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-SQL port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
7. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** drop-down list box, do one of the following:

📌 Note
If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication .

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
User name	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive. 📌 Note If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.

8. Click the **URI Properties** tab, and then specify values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
9. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations you want to target. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

📌 Note
If you selected an existing WCF-SQL send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

10. Click **OK**.

Specifying the Connection URI from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you can specify the connection URI as part of the WCF-Custom or a WCF-SQL port configuration.

To specify the connection URI for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Address (URI)** text box, specify the connection URI to connect to SQL Server. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
6. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab. From the **Binding Type** drop-down list, select **sqlBinding**.
7. If you are creating a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.
8. If you are creating a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab, and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.
9. Click **OK**.

To specify the connection URI for the WCF-SQL port

1. Start the BizTalk Server Administration console.

2. Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.

 **Note**

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the transport properties dialog box, click the **General** tab.
6. Click the **Configure** button and provide values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
7. In the transport properties dialog box, click the **Binding** tab and specify values for binding properties.

 **Note**

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

8. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:
 - Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

 **Note**

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

9. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:
 - Select **User account** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

 **Note**

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

10. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying Credentials for SQL Server

The SQL adapter requires the adapter clients to provide client credentials. The adapter uses these credentials to authenticate the user with SQL Server and to establish a connection.

Adapter clients can provide the client credentials both when using Visual Studio and when using the BizTalk Server Administration console. When using Visual Studio, credentials are required to generate the metadata. When using the BizTalk Server Administration console, credentials are required to perform operations on SQL Server.

This section provides information about specifying client credentials in Visual Studio and the BizTalk Server Administration console.

Specifying Credentials from Visual Studio

From Visual Studio, you can specify the credentials using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify credentials using Consume Adapter Service Add-in

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **sqlBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, do one of the following:

<p> Note</p> <p>If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication.</p>
--

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
Username	<p>Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive.</p> <p> Note</p> <p>If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.</p>

6. Click **OK**.

To specify credentials using Add Adapter Metadata Wizard

1. Right-click your BizTalk project, point to **Add**, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

- Click **Add**. The Add Adapter Metadata Wizard opens.
- In the Add Adapter Wizard, select **WCF-SQL**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

<p>Important</p> <p>If you already have a WCF-SQL port configured in BizTalk, select the port from the Port list.</p>

- Click **Next**.
- In the **Consume Adapter Service** dialog box, from the **Select a binding** list, select **sqlBinding**, and then click **Configure**.
- In the **Configure Adapter** dialog box, click the **Security** tab, and from the **Client credential type** list, do one of the following:

<p>Note</p> <p>If you are connecting to SQL Server using Windows Authentication, the Windows user with which you are logged in must be added to SQL Server as described in Connecting to SQL Server Using Windows Authentication.</p>
--

Click this	To do this
None	Connect to SQL Server by using Windows authentication.
Windows	Connect to SQL Server by using Windows authentication.
User name	Specify the user name and password to connect to SQL Server by specifying credentials for a user defined in SQL Server database. Note that the user name and password are case-sensitive.
	<p>Note</p> <p>If you leave the User name and Password fields as blank, the adapter connects to SQL Server using Windows authentication.</p>

- Click **OK**.

Specifying Credentials from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you can specify the credentials as part of the WCF-Custom or WCF-SQL port configuration.

To specify credentials for the WCF-Custom port

- Start the BizTalk Server Administration console.
- In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
- In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

<p>Note</p> <p>To see the location properties dialog box for a receive port, click the Receive Location tab on the left pane of the port properties dialog box, and then click New.</p>
--

4. If you are creating a send port, in the **WCF-Custom Transport Properties** dialog box, click the **Credentials** tab, and do one of the following:

- Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

5. If you are creating a receive port, in the **WCF-Custom Transport Properties** dialog box, click the **Other** tab, and do one of the following:

- Select the **User account** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Get credentials from affiliate application** option, and specify an affiliate SSO application.

6. Click **OK**.

To specify credentials for the WCF-SQL port

1. Start the BizTalk Server Administration console.
2. Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. If you are creating a send port, in the transport properties dialog box, click the **Credentials** tab, and do one of the following:

- Select the **Do not use Single Sign-On** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and specify an affiliate Enterprise Single Sign-on (SSO) application.

6. If you are creating a receive port, in the transport properties dialog box, click the **Other** tab, and do one of the following:

- Select the **User account** option, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

 **Note**

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Get credentials from affiliate application** option, and specify an affiliate SSO application.

7. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying Binding Properties

The SQL adapter surfaces several binding properties that enable you to control some of its behavioral characteristics. This section provides information about setting the binding properties from Visual Studio and from the BizTalk Server Administration console. From Visual Studio, you must specify the binding properties to generate schema for specific operations. From BizTalk Server, you must specify the binding properties as part of the send or receive port for sending or receiving messages from SQL Server.

For information about the binding properties, including a list of binding properties for the SQL adapter, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Specifying Binding Properties from Visual Studio

From Visual Studio, you can specify the binding properties using the Consume Adapter Service Add-in or the Add Adapter Metadata Wizard.

To specify binding properties using Consume Adapter Service Add-in

1. Right-click your BizTalk project, and then select **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Consume Adapter Service .
Templates	Click Consume Adapter Service .

3. To start the **Consume Adapter Service** dialog box, click **Add**.
4. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
5. In the **Configure Adapter** dialog box, click the **Binding Properties** tab, and then specify the different binding properties.
6. Click **OK**.

To specify the binding properties using Add Adapter Metadata Wizard

1. Right-click the BizTalk project, point to **Add**, and then click **Add Generated Items**.
2. In the **Add Generated Items** dialog box, do the following:

Use this	To do this
Categories	Click Add Adapter .
Templates	Click Add Adapter Metadata .

3. Click **Add**. The Add Adapter Metadata Wizard opens.
4. In the Add Adapter Wizard, select **WCF-SQL**. Select the computer on which BizTalk Server is installed and the name of the BizTalk database.

◆ Important
If you already have a WCF-SQL port configured in BizTalk, select the port from the Port list.

5. Click **Next**.
6. In the **Consume Adapter Service** dialog box, from the **Select a binding** drop-down list, select **sqlBinding**, and then click **Configure**.
7. Click the **Binding Properties** tab, and then specify values for the binding properties, if any, required by the operations

you want to target. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note

If you selected an existing WCF-SQL send port, you need not specify the binding properties. The binding properties are picked from the send port configuration. However, you may choose to specify the binding properties that are required at design-time, if any. In such case, the new values for binding properties will be used at design-time while generating the metadata. However, at run-time the values specified for binding properties in the send port configuration will be applicable.

8. Click **OK**.

Specifying Binding Properties from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you can specify the binding properties as part of the WCF-Custom or WCF-SQL port configuration.

To specify binding properties for WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

4. In the **WCF-Custom Transport Properties** dialog box, click the **Binding** tab.
5. From the **Binding Type** list, select **sqlBinding**.
6. In the **Configuration** box, specify the values for the different binding properties, and then click **OK**.

To specify the binding properties for the WCF-SQL port

1. Start the BizTalk Server Administration console.
2. Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then expand the application under which you want to create a port, and click **Send Ports** or **Receive Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.

Note

To see the location properties dialog box for a receive port, click the **Receive Location** tab on the left pane of the port properties dialog box, and then click **New**.

5. In the transport properties dialog box, click the **Binding** tab and specify values for binding properties.

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

6. Click **OK**.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Specifying SOAP Action

To perform any operation on SQL Server using the WCF-based SQL adapter, you must specify a SOAP action. The SOAP action communicates to the adapter what action should be performed. You can specify the SOAP action either from Visual Studio or from the BizTalk Server Administration console. However, if you specify the SOAP action from both locations, the action you specified from Visual Studio will be overridden.

For more information about specifying SOAP action, see <http://go.microsoft.com/fwlink/?LinkId=102063>.

Specifying SOAP Action from Visual Studio

From Visual Studio, you must specify the SOAP action as part of the orchestration by using an **Expression** shape.

To specify SOAP action from Visual Studio

1. In the BizTalk orchestration, include an **Expression** shape by dragging it from the **BizTalk Orchestration** toolbox.
2. Double-click the **Expression** shape to open BizTalk Expression Editor.
3. Specify the action in BizTalk Expression Editor. For example:

```
OutboundMessage(WCF.Action)="TableOp/Insert/dbo/Employee"
```

For more information about the **Expression** shape and BizTalk Expression Editor, see <http://go.microsoft.com/fwlink/?LinkId=102064>.

Specifying SOAP Action from the BizTalk Server Administration Console

From the BizTalk Server Administration console, you can specify the SOAP action as part of the WCF-Custom or WCF-SQL port configuration.

To specify a SOAP action for the WCF-Custom port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
3. In the port properties dialog box, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
4. In the **WCF-Custom Transport Properties** dialog box, click the **General** tab.
5. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:

- **By using the single action format.** Use this format if the WCF-Custom port sends and receive messages for a single operation. For example:

```
TableOp/Insert/dbo/Employee
```

- **By using the action mapping format.** Use this format if a single WCF-Custom port sends and receives messages for more than one operation. For example, if a single WCF-Custom port sends and receives messages for Op1 (to insert records in the Employee table) and Op2 (to update records in the Employee table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>
  <Operation Name="Op1" Action="TableOp/Insert/dbo/Employee" />
  <Operation Name="Op2" Action="TableOp/Update/dbo/Employee" />
</BtsActionMapping>
```

The action mapping approach provides greater flexibility in terms of specifying a set of actions, and hence enabling messages that belong to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about the action format for each operation, see individual topics under [Technical Reference](#).

To specify a SOAP action for the WCF-SQL port

1. Start the BizTalk Server Administration console.
2. Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, then expand **Applications**, and then click **Send Ports**. In the right pane, you can choose to create a port or select an existing port.
4. In the port properties dialog box, from the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.
5. In the transport properties dialog box, click the **General** tab.
6. In the **Action** text box, specify the SOAP action for the operation. You can specify the action in the following ways:
 - **By using the single action format.** Use this format if the WCF-SQL port sends and receive messages for a single operation. For example:

```
TableOp/Insert/dbo/Employee
```

- **By using the action mapping format.** Use this format if a single WCF-SQL port sends and receives messages for more than one operation. For example, if a single WCF-SQL port sends and receives messages for Op1 (to insert records in the Employee table) and Op2 (to update records in the Employee table), the SOAP action can be specified in the following manner:

```
<BtsActionMapping>  
  <Operation Name="Op1" Action="TableOp/Insert/dbo/Employee" />  
  <Operation Name="Op2" Action="TableOp/Update/dbo/Employee" />  
</BtsActionMapping>
```

The action mapping approach provides greater flexibility in terms of specifying a set of actions, and hence enabling messages that belong to different action types to flow through the same port.

The format for the SOAP action is different for each operation. For more information about the action format for each operation, see individual topics under [Technical Reference](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Manually Configuring a Physical Port Binding to the SQL Adapter

This section provides information about configuring the Microsoft BizTalk Adapter for SQL Server as a WCF custom binding or as a WCF-SQL port by using the BizTalk Server Administration console. After deploying the adapter, you will be able to send and receive messages from SQL Server by using the BizTalk Server Administration console. The steps for deploying the adapter vary depending on:

- The direction of communication between BizTalk Server and the SQL adapter. You may choose to configure a send, receive, or a send-receive port. Your choices are summarized in the following table.

Port direction	Communication pattern	Direction of communication to choose from
Send	One-way	I will always be sending messages on this port.
Receive	One-way	I will always be receiving messages on this port.
Send-receive	Request-response	I will be sending a request and receiving a response.

Note

Two-way receive ports are not supported by the SQL adapter.

For more information, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

- Whether the adapter sends messages to the SQL Server (outbound operations) or receives messages from SQL Server (inbound operations). Depending on whether you want to send or receive messages, you will create a send or receive port, respectively.

Note

You can also configure the send or receive ports by importing a binding configuration file that is created by the Consumer Adapter Service Add-in as part of metadata generation. For instructions on configuring ports using this binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

In This Section

- [Configuring a Port Using the WCF-Custom Adapter](#)
- [Configuring a Port Using the WCF-SQL Adapter](#)

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring a Port Using the WCF-Custom Adapter

This topic provides instructions on how to configure WCF-Custom send and receive ports to perform outbound and inbound operations on SQL Server using the SQL adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to SQL Server

Perform the following steps to configure a WCF-Custom send port for sending messages to SQL Server using the BizTalk Server Administration console.

To deploy adapters for sending messages to SQL Server

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the SQL adapter.
4. Right-click **Send Ports**, point to **New**, and then point to the type of port you want to configure depending on the mode of communication between BizTalk Server and SQL Server.
5. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
6. From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
7. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for SQL Server. For more information about the connection URI, see [SQL Server Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on a table in a SQL Server database is:

TableOp/Insert/dbo/Employee

Note

Employee is the name of a table in SQL Server database.

- c. Click the **Binding** tab, and from the **Binding Type** list, select **sqlBinding**. You can specify the different binding properties exposed by the SQL adapter. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and then specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and then specify an affiliate Enterprise Single Sign-on (SSO) application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

- To return to the **Send Port Properties** dialog box, click **OK**.
- From the **Send handler** list, select **BizTalkServerApplication**.
 - If you chose **Static One-Way Send Port** in step 4, specify a send pipeline. From the **Send pipeline** list, select the pipeline that corresponds to XMLTransmit.
 - If you chose **Static Solicit-Response Port** in step 4, specify send and receive pipelines.
 - From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - Click **OK**.

Deploying Adapters for Receiving Messages from SQL Server

Perform the following steps to configure a WCF-Custom receive port for receiving messages from SQL Server using the BizTalk Server Administration console.

To deploy adapters for receiving messages from SQL Server

- Start the BizTalk Server Administration console.
- In the console tree, expand **BizTalk Group**, and then expand **Applications**.
- Expand the application under which you want to deploy the SQL adapter.
- Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and SQL Server.
- In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
- On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
- In the **Receive Location Properties** dialog box, do the following:
 - Specify a name for the receive location.
 - From the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
- In the **WCF-Custom Transport Properties** dialog box, do the following:
 - Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for SQL Server. For more information about the connection URI, see [SQL Server Connection URI](#).
 - Click the **Binding** tab, and from the **Binding Type** drop-down list, select **sqlBinding**. You can specify the different binding properties exposed by the SQL adapter. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
 - Click the **Behavior** tab to set the transaction isolation level. For more information about setting transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).
 - Click the **Other** tab, and do one of the following:

- Select **User account**, and specify the user name and password to connect to SQL Server. Note that the user

name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

- e. To return to the **Receive Location Properties** dialog box, click **OK**.

9. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.
10. If you chose **One-way Receive Port** in step 4, specify a receive pipeline. From the **Receive pipeline** list, select the pipeline corresponding to XMLReceive.
11. If you chose **Request Response Receive Port** in step 4, specify send and receive pipelines.
 - a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
12. In the **Receive Location Properties** dialog box, click **OK**.
13. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Other Resources

[Manually Configuring a Physical Port Binding to the SQL Adapter](#)

Configuring a Port Using the WCF-SQL Adapter

This topic provides instructions on how to configure WCF-SQL send and receive ports to perform outbound and inbound operations on SQL Server using the SQL adapter.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

Deploying Adapters for Sending Messages to SQL Server

Perform the following steps to configure a WCF-SQL send port for sending messages to SQL Server using the BizTalk Server Administration console.

To deploy adapters for sending messages to SQL Server

1. Start the BizTalk Server Administration console.
2. Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
3. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
4. Expand the application under which you want to deploy the SQL adapter.
5. Right-click **Send Ports**, point to **New**, and then point to the type of port you want to configure depending on the mode of communication between BizTalk Server and SQL Server.
6. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
7. From the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.
8. In the transport properties dialog box, do the following:
 - a. Click the **General** tab, click the **Configure** button, and provide values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the Insert operation on a table in a SQL Server database is:

TableOp/Insert/dbo/Employee

Note

Employee is the name of a table in SQL Server database.

- c. Click the **Binding** tab and specify values for binding properties exposed by the SQL adapter. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and then specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select the **Use Single Sign-On** option, and then specify an affiliate Enterprise Single Sign-on (SSO) application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

- To return to the **Send Port Properties** dialog box, click **OK**.
- From the **Send handler** list, select **BizTalkServerApplication**.
 - If you chose to create **Static One-Way Send Port** in step 5, specify a send pipeline. From the **Send pipeline** list, select the pipeline that corresponds to XMLTransmit.
 - If you chose to create **Static Solicit-Response Port** in step 5, specify send and receive pipelines.
 - From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 - From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - Click **OK**.

Deploying Adapters for Receiving Messages from SQL Server

Perform the following steps to configure a WCF-SQL receive port for receiving messages from SQL Server using the BizTalk Server Administration console.

To deploy adapters for receiving messages from SQL Server

- Start the BizTalk Server Administration console.
- Add the WCF-SQL adapter to the BizTalk Server Administration console. For instructions, see [Adding the SQL Adapter to BizTalk Server Administration Console](#).
- In the console tree, expand **BizTalk Group**, and then expand **Applications**.
- Expand the application under which you want to deploy the SQL adapter.
- Right-click **Receive Ports**, point to **New**, and click **One-way Receive Port** or **Request Response Receive Port**, depending on the mode of communication between BizTalk Server and SQL Server.
- In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
- On the **Receive Locations** tab, click **New**. The **Receive Location Properties** dialog box appears.
- In the **Receive Location Properties** dialog box, do the following:
 - Specify a name for the receive location.
 - From the **Type** drop-down list, select the WCF-SQL adapter you added earlier, and then click **Configure**.
- In the transport properties dialog box, do the following:
 - Click the **General** tab, click the **Configure** button, and provide values for the connection parameters. For more information about the connection URI for the SQL adapter, see [SQL Server Connection URI](#).
 - Click the **Binding** tab and specify values for binding properties exposed by the SQL adapter. For more information

about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note

The binding properties are displayed based on whether you are configuring a send port or a receive port. For example, binding properties related to notifications are not available while configuring a send port because notifications are inbound operations and require a receive port configuration.

- c. Click the **Behavior** tab to set the transaction isolation level. For more information about setting transaction isolation level, see [Configure Transaction Isolation Level and Transaction Timeout](#).
- d. Click the **Other** tab, and do one of the following:
 - Select **User account**, and specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

Note

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password. Before you do this, the Windows user with which you are logged in must be added to SQL Server as described in [Connecting to SQL Server Using Windows Authentication](#).

- Select **Get credentials from affiliate application** option, and specify an affiliate SSO application.

For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).

- e. To return to the **Receive Location Properties** dialog box, click **OK**.
10. From the **Receive handler** drop-down list, select **BizTalkServerApplication**.
 11. If you chose to create **One-way Receive Port** in step 5, specify a receive pipeline. From the **Receive pipeline** list, select the pipeline corresponding to XMLReceive.
 12. If you chose to create **Request Response Receive Port** in step 5, specify send and receive pipelines.
 - a. From the **Receive pipeline** drop-down list, select the pipeline that corresponds to XMLReceive.
 - b. From the **Send pipeline** drop-down list, select the pipeline that corresponds to XMLTransmit.
 13. In the **Receive Location Properties** dialog box, click **OK**.
 14. In the **Receive Port Properties** dialog box, click **OK**.

See Also

Other Resources

[Manually Configuring a Physical Port Binding to the SQL Adapter](#)

Configuring a Physical Port Binding Using a Port Binding File

When you use the Consume Adapter Service BizTalk Project Add-in to generate metadata for a SQL Server artifact, other than the schema files, the Consume Adapter Service Add-in also generates a port binding file. You can import this binding file into your BizTalk application to create a physical send or receive port. For instructions on importing binding files, see [Importing Bindings](#). If you import this binding file, you do not have to manually create a physical send or receive port.

Important

While using the Consume Adapter Service Add-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the binding file. You must manually add the binding property and its value in the binding file, if required.

Creating a port using the port binding file always creates a two-way send port or a one-way receive port. If you want to create a one-way send port, you can create it manually by following the procedure mentioned in [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Or, you can follow the workarounds documented in this topic to modify the port binding file to create one-way send ports.

Note

For inbound operations, the port binding file will always create a one-way receive port. This is because the SQL adapter only supports one-way receive port for inbound operations.

Important

Using the Add Adapter Metadata Wizard does not create a port binding file using which you can create a WCF-SQL port. However, you could make some changes to the port binding file generated by the Consume Adapter Service Add-in and use it to create a WCF-SQL port. For more information, see [Configuring a WCF-SQL Port Using the Port Binding File Generated Using Consume Adapter Service Add-in](#).

Following are some key points that you must understand with respect to the binding file that the Consume Adapter Service Add-in generates:

- The files are created with a specific naming convention. If you generated metadata for outbound operations, that is, to send messages to SQL Server, the name of the file is `WcfSendPort_SqlAdapterBinding_Custom.bindinginfo.xml`.

If you generated metadata for inbound operations, that is, to receive messages from SQL Server, the name of the file is `WcfReceivePort_SqlAdapterBinding_Custom.bindinginfo.xml`.

- The file contains information about the binding configuration, the binding type, the endpoint URI, and the port action based on the operations for which metadata was generated. When you import this binding file into your BizTalk application to create a port, all the relevant information required to configure a physical port is automatically set on the port.

Important

By default, the action on the send port is mapped to the operation name for which you generated metadata. For example, if you generate metadata for the Insert operation on the Customer table, the action on the port is set to `<Operation Name="Insert" Action="TableOp/Insert/dbo/CustomerTable" />`. However, the operation name on the logical send port you create in the BizTalk orchestration might not be the same. You must ensure that the operation name in the logical send port (in the BizTalk orchestration) and the physical send port (in BizTalk Server Administration console) are the same. If not, you will receive an error while sending messages to SQL Server through the send port.

- You only need to provide the credentials for the port to connect to SQL Server. Although the binding file does retain the user name with which to connect, for security reasons the binding file does not contain the password.

- When you import the binding file, you might get a dialog message informing that the BizTalk application name in the binding file does not match the application name to which you are importing the binding file. You can safely ignore this message and continue.
- The binding file also contains names of ports and receive locations. If the BizTalk application to which you are importing the binding file creates a port or a receive location that has the same name as an already existing port in the same BizTalk application, you will get an error. You must manually edit the binding file to specify a unique name for the ports or receive locations.
- The binding file also contains information about the connection URI. If the binding file creates a receive location that has the same receive URI as an already existing receive location in the same BizTalk application, you will get an error. You must manually edit the binding file to specify a unique URI.
- By default, the port binding file for outbound operations always contains definitions for a two-way send port. When you import this file in a BizTalk application, it creates a two-way send port. However, you may have an orchestration that has a one-way send port. So, when you configure such an orchestration and use the port created by importing the binding file, the port is not available in the list. This happens because the logical port you created as part of the orchestration is a one-way port while the physical port created in the orchestration is a two-way port. In such cases, you can edit the binding file to make the following changes:

For this	Do this
To edit the port binding file to configure a one-way send port	<ol style="list-style-type: none"> In the following excerpt, change the value of the IsTwoWay property to false. Originally, this is set to true. <div data-bbox="687 958 1498 1081" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><SendPort Name="port_name" IsStatic="true" IsTwoWay="false" BindingOption="0"></pre> </div> Comment out the following excerpts: <div data-bbox="687 1171 1498 1503" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipeline Name="Microsoft.BizTalk.DefaultPipelines.XMLReceive" FullyQualifiedName="Microsoft.BizTalk.DefaultPipelines.XMLReceive, Microsoft.BizTalk.DefaultPipelines, Version=3.0.1.0, Culture=neutral, PublicKeyToken=token" Type="1" TrackingOption="None" Description=""/></pre> </div> <div data-bbox="687 1518 1498 1608" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre><ReceivePipelineData xsi:nil="true" /></pre> </div>

◆ Important
For inbound operations, the port binding file will always create a one-way receive port. This is because the SQL adapter only supports one-way receive port for inbound operations.

Configuring a WCF-SQL Port Using the Port Binding File Generated Using Consume Adapter Service Add-in

The Consume Adapter Service Add-in also creates a port binding file that you can import in BizTalk Server Administration console. You can use the same port binding file to also create the BizTalk WCF-SQL port in BizTalk Server Administration console. However, before creating WCF-SQL port you must perform the following tasks to modify the port binding file.

1. Open the port binding file in a text editor.

2. Search and replace "WCF-Custom" with the name with which you added the WCF-SQL adapter in BizTalk Server Administration console. For example, if you added the WCF-SQL adapter as "SQLAdapter", replace "WCF-Custom" with "SQLAdapter".
3. Search for the "ConfigurationClsid" attribute, and replace the existing value of the attribute with "59B35D03-6A06-4734-A249-EF561254ECF7".
4. Save and close the binding file.
5. Import the binding file in BizTalk Server Administration console. For instructions on how to import the binding file, see [Importing Bindings](#).

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Configuring Dynamic Ports

In BizTalk Server, you can configure dynamic ports for a WCF-Custom adapter. Because the SQL adapter is a WCF-based adapter, you can dynamically configure a port for the SQL adapter by using message context properties.

For the SQL adapter, the URI, action, and binding may be determined from a property on an incoming message, and then specified in the **Expression** shape, as shown in the following example:

```
Request2=Request1;
Request2(WCF.Action)="TableOp/Insert/dbo/CustomersTable";
Request2(WCF.BindingType)="sqlBinding";
Request2(WCF.UserName)="myuser";
Request2(WCF.Password)="mypass";
SendPort(Microsoft.XLANGs.BaseTypes.Address)="mssql://sql_server/my_instance/my_database";
SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="WCF-Custom";
```

Note

If you are using a WCF-SQL adapter in BizTalk Server Administration console, you can also specify the transport type as `SendPort(Microsoft.XLANGs.BaseTypes.TransportType)="SQLAdapter"`, where **SQLAdapter** is the name with which you added the WCF-SQL adapter in BizTalk Server Administration console.

In the preceding example,

- Request2 message is being created from Request1 message. Both messages map to an operation schema, which is generated using the Consume Adapter Service Add-in.
- SendPort is the name of the logical send port in the BizTalk orchestration.

The **Expression** shape is part of the BizTalk orchestration. Deploying the orchestration also creates a WCF-Custom send port.

For more information about configuring dynamic ports, see "Configuring Dynamic Send Ports Using WCF Adapters Context Properties" at <http://go.microsoft.com/fwlink/?LinkId=106735>.

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Reusing Adapter Bindings

A binding creates a mapping between a logical endpoint (such as an orchestration port or a role link) and a physical endpoint (such as a send and receive port). This enables communication between different components of a BizTalk business solution. You can create bindings by using the BizTalk Server Administration console.

What is a Binding File?

A binding file is an XML file that contains binding information for each BizTalk orchestration in the scope of a BizTalk assembly, application, or group. The binding file describes:

- The host to which each orchestration is bound.
- The trust level of the host.
- The settings for each send port, receive port, receive location, and party that has been configured.

You can generate binding files and then apply the bindings that they contain to an assembly, application, or group. This prevents having to manually configure bindings in different deployment environments and speeds up application deployment.

A binding file is not automatically generated for a BizTalk assembly, application, or group. However, you can generate a binding file by exporting bindings. Similarly, you can then import the binding file into an application or group. This section provides instructions on how to import and export bindings.

For more information about bindings and about binding files, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

In This Section

- [Exporting Bindings](#)
- [Importing Bindings](#)

See Also

Tasks

[Building Blocks for Developing BizTalk Applications](#)

Exporting Bindings

This section describes how to export bindings for a BizTalk application into an XML file. You can then import the bindings from the XML file into another BizTalk application. Importing bindings overwrites any existing bindings of the same name in the application. You can also add bindings to an application, which does not overwrite existing bindings. The bindings that you add do not take effect until you import the application.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To export bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application whose bindings you want to export, point to **Export**, and then click **Bindings**.
4. On the **Export Bindings** page, in **Export to file**, type the absolute path of the XML file to which to export the bindings.
Example: C:\Bindings\Application1Bindings.xml.
5. Ensure that **Export all bindings from the current application** is selected.
6. To export all party information for the group, select the **Export Global Party information** check box.
7. Click **OK**.

The bindings are exported into an XML file in the location that you specified.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the **Transport Properties** dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for SQL Server](#).

See Also

Concepts

[Reusing Adapter Bindings](#)

Importing Bindings

This topic describes how to use the BizTalk Server Administration console to import bindings into a BizTalk application from an XML file.

Prerequisites

To perform the procedures in this topic, you must be logged on with an account that is a member of the BizTalk Server Administrators or BizTalk Operators group. For more detailed information about permissions, see the BizTalk Server Help documentation at <http://go.microsoft.com/fwlink/?LinkID=101130>.

To import bindings using the BizTalk Server Administration console

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Right-click the application into which you want to import bindings, point to **Import**, and then click **Bindings**.
4. Click the binding file, and then click **Open**.

The artifacts in the binding file are written to the application. They are displayed in appropriate folders of the application. For example, the send ports that are imported as part of the bindings are displayed under the Send Ports folder.

Note

For security reasons, when you export a binding file, BizTalk Server removes the passwords for the bindings from the file. After importing the bindings, you must reconfigure passwords for send ports and receive locations before they will function. You configure passwords in the **Transport Properties** dialog box of the BizTalk Server Administration console for the send port or receive location. For information about specifying user name and passwords, see [Specifying Credentials for SQL Server](#).

See Also

Concepts

[Reusing Adapter Bindings](#)

Performing Basic Insert, Update, Delete, and Select Operations by Using BizTalk Server

The Microsoft BizTalk Adapter for SQL Server surfaces a set of standard operations on SQL Server database tables and views. These are called data manipulation language (DML) operations. By using DML operations, you can perform simple Insert, Update, Select, and Delete operations on tables and views. For more information about how the adapter supports these operations, see [Performing Basic Insert, Update, Delete, and Select Operations on Tables and Views](#). For information about the structure of the SOAP message for these operations, see [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#).

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

How to Perform Basic Operations on a SQL Server Database

Performing an operation on a SQL Server database by using SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform Insert, Update, Delete, or Select operations on tables and views in SQL Server, these tasks are:

1. Create a BizTalk project, and generate schema for the operation you want to invoke on a SQL Server database table or view.
2. Create messages in the BizTalk project for sending and receiving messages from the SQL Server database.
3. Create an orchestration to invoke the operation on the SQL Server database table or view.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based On This Topic

A sample, SelectTable, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

This topic demonstrates how to perform basic DML operations by selecting records from the EMPLOYEE table in the SQL Server database. Run the scripts provided with the samples to create the EMPLOYEE table. For more information about the samples, see [Samples](#).

To demonstrate how to select records, schema is generated for the Select operation for the EMPLOYEE table. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schemas.

Important

If you are generating metadata for operations on a table that has columns of user-defined types (UDTs), make sure the respective assemblies of the UDTs are available at the same location as the Visual Studio executable, devenv.exe. The executable is typically available at `<installation drive>:\Program Files\Microsoft Visual Studio <version>\Common7\IDE`. In this example, the EMPLOYEE table has a UDT (Point) column. Make sure you copy the respective assembly at the same location as the Visual Studio executable.

For information about how to create a UDT, see <http://go.microsoft.com/fwlink/?LinkId=138066>. For information about how to register a UDT in SQL Server, see <http://go.microsoft.com/fwlink/?LinkId=138067>.

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

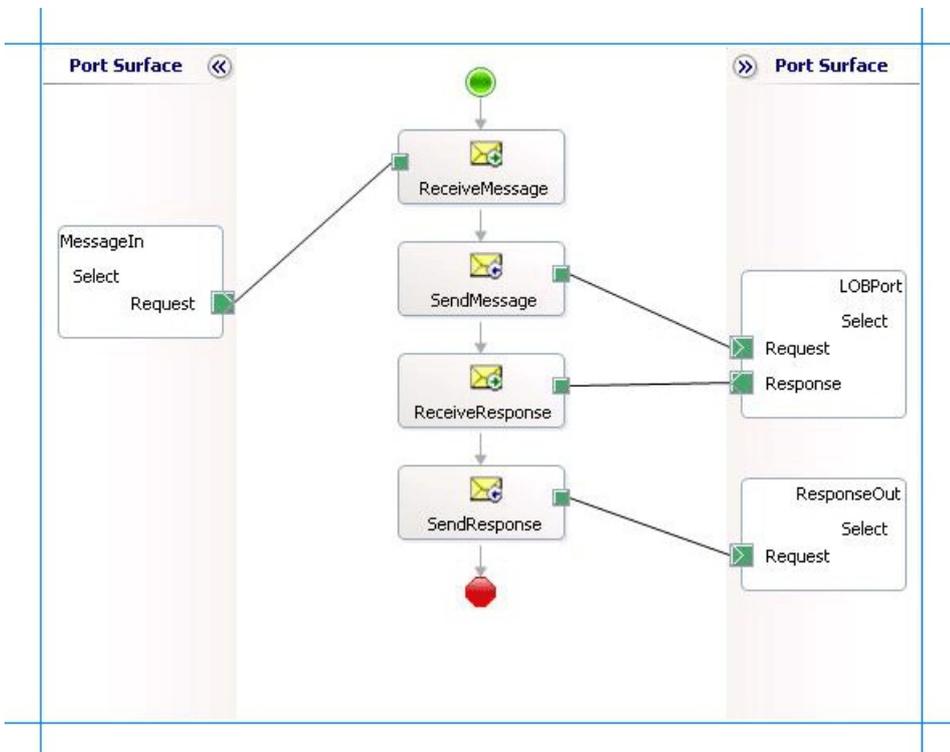
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>SelectTable.TableOperation_dbo_Employee.Select</i> , where <i>SelectTable</i> is the name of your BizTalk project. <i>TableOperation_dbo_Employee</i> is the schema generated for the <i>Select</i> operation on the <i>EMPLOYEE</i> table.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>SelectTable.TableOperation_dbo_Employee.SelectResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and to receive responses, respectively. A sample orchestration for the Select operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> Set Identifier to <i>MessageIn</i> Set Type to <i>MessageInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.Select.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Select.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Select.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.Select.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console.

For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
- Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for selecting records from a SQL Server database table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to the SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Important

If you are performing operations on a table that has columns of user-defined types (UDTs), make sure the respective assemblies of the UDTs are available at the same location as the BizTalk Server executable, `btsntsvc.exe`. The executable is typically available at `<installation drive>:\Program Files\Microsoft BizTalk Server <version>`. In this example, the EMPLOYEE table has a UDT (Point) column. Make sure you copy the respective assembly at the same location as the BizTalk Server executable.

For information about how to create a UDT, see <http://go.microsoft.com/fwlink/?LinkId=138066>. For information about how to register a UDT in SQL Server, see <http://go.microsoft.com/fwlink/?LinkId=196964>.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the Select operation that you generated earlier. For example, the request message to select all records from the EMPLOYEE table is:

```
<Select xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <Columns>*</Columns>
  <Query>where Employee_ID=10001</Query>
</Select>
```

This request message will retrieve records from the Employee table that satisfy the condition specified in the `<Query>` element. If you want to retrieve specific columns from the table, you must specify them in the `<Columns>` element, separated by comma, in the same sequence as they appear in the table definition. If you do not want to specify a condition to retrieve data, leave the `<Query>` element blank. See [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#)

for more information about the request message schema for performing basic DML operations on SQL Server database tables and views using the SQL adapter.

The orchestration consumes the message and sends it to SQL Server database. The response from SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<SelectResponse xmlns="mssql://Microsoft.LobServices.Sql/2008/01/TVOp/dbo/Employee">
  <SelectResult>
    <Employee xmlns="mssql://Microsoft.LobServices.Sql/2008/01/Types/Tables/dbo">
      <Employee_ID>10001</Employee_ID>
      <Name>John</Name>
      <DOJ>1983-12-31T00:00:00Z</DOJ>
      <Designation>Manager</Designation>
      <Job_Description>Management</Job_Description>
      <Photo>EjRVYzRFVQ==</Photo>
      <Rating>1,2</Rating>
      <Salary>100000.00</Salary>
      <Last_Modified>AAAAAAAAD6I=</Last_Modified>
    </Employee>
  </SelectResult>
</SelectResponse>
```

Best Practices

- After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).
- If you are inserting, updating, or deleting large volumes of data make sure you set the right timeout values for the WCF adapter and for the MSDTC transaction. For more information, see "The adapter fails to insert, update, or delete large volumes of data in a single operation using BizTalk Server" issue in [Troubleshooting Operational Issues](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Operations on Tables and Views with Large Data Types by Using BizTalk Server

The SQL adapter enables adapter clients to read and update data in columns of large data types, that is, `varchar(max)`, `nvarchar(max)`, or `varbinary(max)`. To read data from such columns, adapter clients can use the `Select` operation. To insert or update data into such columns, the adapter exposes a `Set<column_name>` operation, where `<column_name>` is the name of the column of type `varchar(max)`, `nvarchar(max)`, or `varbinary(max)`.

Additionally, in SQL Server 2008, you can have the `varbinary(max)` column store unstructured data such as text documents and images. Such unstructured data is called FILESTREAM data. FILESTREAM data can be stored as files on the file system. The SQL adapter enables the client to enter FILESTREAM data into columns of type `varbinary(max)`. For more information about FILESTREAM storage, see <http://go.microsoft.com/fwlink/?LinkId=122485>.

This topic provides information about certain tasks you must perform on the computer running SQL Server and the computer running the adapter client to be able to insert or update FILESTREAM data. This topic also provides instructions on performing `Set<column_name>` operations to insert FILESTREAM data.

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

Prerequisites

You must perform the following tasks on the computer running SQL Server 2008 and the computer running the adapter client.

Note

If you are performing `Set<column_name>` operations on SQL Server 2005, you do not need to perform these tasks. You can directly start with the next section. This is because FILESTREAM is not supported on SQL Server 2005.

• On the computer running SQL Server 2008

- You must enable FILESTREAM on the SQL Server instance. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122486>.
- You must create a FILESTREAM-enabled database. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122487>.
- You must have a table for storing FILESTREAM data. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122488>.
- You must configure MSDTC on the computer hosting the SQL Server database. For instruction on how to configure MSDTC, see [Configure MSDTC on SQL Server and Adapter Client](#).

• On the computer running the adapter client

- You must have the SQL Client Connectivity SDK installed. You can install the SQL Client Connectivity SDK by running the SQL Server 2008 setup and selecting **SQL Client Connectivity SDK** in the **Feature Selection** page of the wizard. The adapter uses the `sqlncli10.dll`, installed with the SQL Client Connectivity SDK, to perform FILESTREAM operations.
- You must configure MSDTC on the computer running the adapter client. For instruction on how to configure MSDTC, see [Configure MSDTC on SQL Server and Adapter Client](#).

After you have completed these tasks, you are all set to insert or update FILESTREAM data in SQL Server 2008 database tables.

How This Topic Demonstrates Operations on Large Data Types

To demonstrate how to perform Set<column_name> operations on tables with large data types, take a table, Records, that has columns Id and Document. The Id column is of type uniqueidentifier and takes a GUID. The Document column is of type VARBINARY(MAX). Assume that the Id column already has a GUID '438B7B4C-5491-409F-BCC1-78817C399EC3'. To update the Document column, the adapter exposes the SetDocument operation.

Note

For SQL Server 2008, to demonstrate FILESTREAM operations, assume that the Document column can store FILESTREAM data.

How to Perform Operations on a SQL Server Database

Performing an operation on a SQL Server database by using SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform operations on tables with large data types, these tasks are:

1. Create a BizTalk project, and generate schema for the Set<column_name> operation. For this topic, generate schema for the **SetDocument** operation for the **Records** table.
2. Create messages in the BizTalk project for sending and receiving messages from the SQL Server database.
3. Create an orchestration to invoke the SetDocument operation on the Records table.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based on This Topic

A sample, FILESTREAMOperation, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

To demonstrate how to update values in columns of large data types, generate schema for the SetDocument operation of the Records table. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

Use this	To do this
-----------------	-------------------

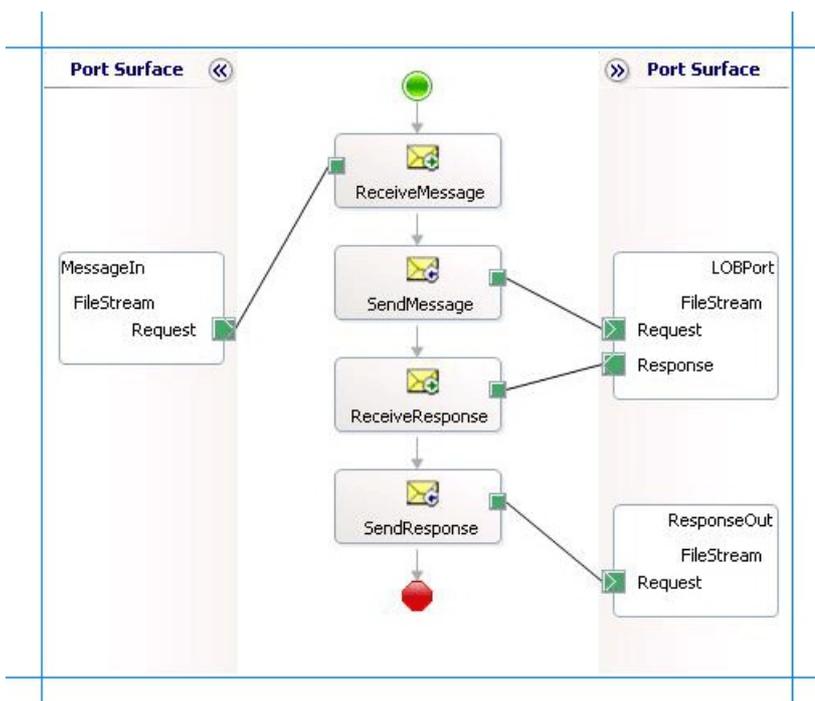
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>SetOperation.TableOperation_dbo_Records.SetDocument</i> , where SetOperation is the name of your BizTalk project. TableOperation_dbo_Records is the schema generated for the SetDocument operation on the Records table.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>SetOperation.TableOperation_dbo_Records.SetDocumentResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and to receive responses, respectively. A sample orchestration for the SetDocument operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>

ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.FileStream.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.FileStream.Request</i>

ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.FileStream.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.FileStream.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

◆ Important

The operation to enter FILESTREAM data must be performed within a transaction. So, make sure the **UseAmbientTransaction** binding property is set to **True** on the WCF-Custom or WCF-SQL send port. For more information about the binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

◆ Important

For performing an operation to insert FILESTREAM data you must always use Windows Authentication to connect to SQL Server on the WCF-Custom or WCF-SQL send port. So, in the **Credentials** tab in the port properties dialog box, select the **Do not use single Sign-On** option, and leave the user name and password blank.

📌 Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for performing the **SetDocument** operation on the **Records** table. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to the SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the SetDocument operation that you generated earlier. For example, the request message to update the Document column is:

```
<SetDocument xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Records">
  <Filter>WHERE Id='438B7B4C-5491-409F-BCC1-78817C399EC3'</Filter>
  <Data>UwBlAHQAdgBfAHYAYQByAGIAaQBuAGEAcgB5AE0AQQBYAA==</Data>
</SetDocument>
```

◆ Important

The `Filter` element must contain the WHERE clause based on which the adapter updates the records. The `Data` element must contain a base64 encoded value that you want to insert into the Document column.

This request message updates the Document column with the specified value. The orchestration consumes the message and sends it to the SQL Server database. The response from the SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from the SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<SetDocumentResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Records" /
>
```

The adapter sends an empty response for the **Set<column_name>** operation.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Executing Stored Procedures in SQL Server by Using BizTalk Server

The Microsoft BizTalk Adapter for SQL Server surfaces the procedures in SQL Server database as operations. Adapter clients can invoke the procedures by using the SQL adapter with BizTalk Server. For more information about how the adapter supports these operations, see [Executing Stored Procedures in SQL Server](#). For information about the structure of the SOAP message for these operations, see [Message Schemas for Procedures and Functions](#).

The SQL adapter also enables adapter clients to invoke:

- Scalar functions in SQL Server database. See [Invoking Scalar Functions in SQL Server by Using BizTalk Server](#).
- Table valued functions in SQL Server database. See [Invoking Table-Valued Functions in SQL Server by Using BizTalk Server](#).

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

How to Invoke Procedures in SQL Server Database

Performing an operation on SQL Server database using the SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke a procedure in SQL Server database, these tasks are:

- Create a BizTalk project, and generate schema for the procedure you want to invoke in SQL Server database.
- Create messages in the BizTalk project for sending and receiving messages to and from SQL Server database.
- Create an orchestration to invoke the procedure in SQL Server database.
- Build and deploy the BizTalk project.
- Configure the BizTalk application by creating physical send and receive ports.
- Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based on This Topic

A sample, `ExecuteStoredProcedure`, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

This topic demonstrates how to invoke a procedure, by invoking the `ADD_EMP_DETAILS` procedure. This procedure is created by running the scripts provided with the samples. The `ADD_EMP_DETAILS` procedure takes certain parameters, inserts a record into the `EMPLOYEE` table, and returns an employee ID for the inserted record. For information about the samples and the SQL scripts, see [Samples](#).

To invoke the `ADD_EMP_DETAILS` procedure, it is necessary to generate schema for the same procedure. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schema.

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration, and link them to schemas that you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

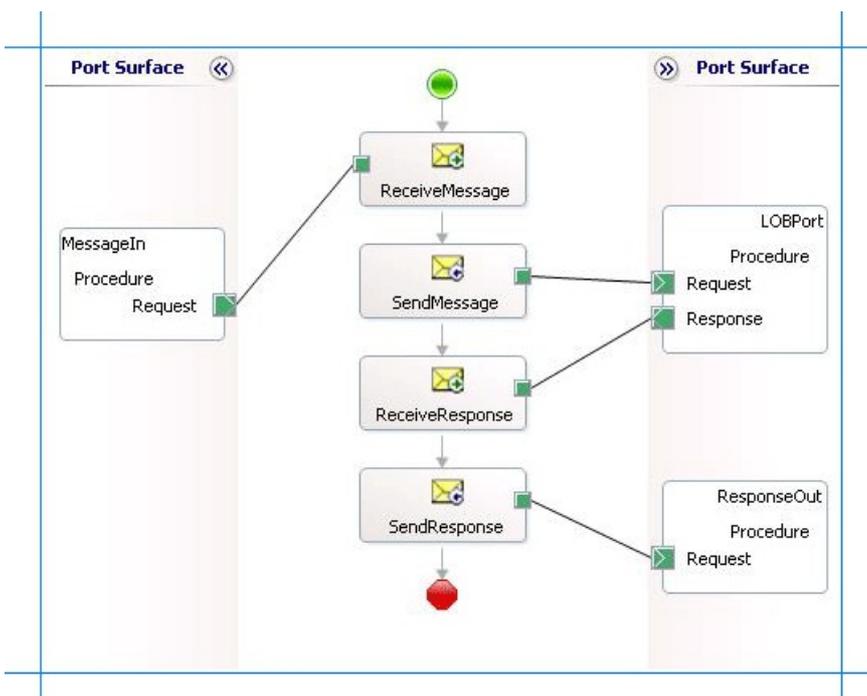
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>ExecProcedure.Procedure_dbo.ADD_EMP_DETAILS</i> , where ExecProcedure is the name of your BizTalk project. Procedure_dbo is the schema generated for invoking the ADD_EMP_DETAILS procedure.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>ExecProcedure.Procedure_dbo.ADD_EMP_DETAILSResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and receive responses, respectively. A sample orchestration for invoking a procedure resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the

names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.Procedure.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.Procedure.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.Procedure.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.Procedure.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking procedures in SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the procedure you generated earlier. For example, the request message to invoke the GET_EMP_DETAILS is:

```
<ADD_EMP_DETAILS xmlns="mssql://Microsoft.LobServices.Sql/2008/01/Procedures/dbo">
  <emp_name>John</emp_name>
  <emp_desig>Developer</emp_desig>
  <salary>100000</salary>
</ADD_EMP_DETAILS>
```

See [Message Schemas for Procedures and Functions](#) for more information about the request message schema for invoking procedures in SQL Server database using the SQL adapter.

The orchestration consumes the message and sends it to SQL Server database. The response from SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<ADD_EMP_DETAILSResponse xmlns="mssql://Microsoft.LobServices.Sql/2008/01/Procedures/dbo">
  <ADD_EMP_DETAILSResult>
    <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data">
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" name="Employee_ID" type="xs:int" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
      <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
        <NewDataSet xmlns="">
          <NewTable>
            <Employee_ID>10001</Employee_ID>
          </NewTable>
        </NewDataSet>
      </diffgr:diffgram>
    </DataSet>
  </ADD_EMP_DETAILSResult>
  <ReturnValue>0</ReturnValue>
</ADD_EMP_DETAILSResponse>
```

In the above response the <Employee_ID> element contains the employee ID for the inserted record.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the

binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Executing Stored Procedures With a Single XML Parameter

This topic was last updated on: June 05, 2009

Executing a stored procedure that takes a single parameter is similar to executing any other stored procedure as described in [Executing Stored Procedures in SQL Server by Using BizTalk Server](#). However, for the approach described in the preceding link, you need to generate metadata for the stored procedure at design time and create an orchestration to invoke the procedure at run time.

Consider a scenario where you just want to pass one single value to a stored procedure without doing any processing on that value. In such cases, you do not want the overhead of generating metadata, creating an orchestration, deploying the orchestration, and executing the operation. Rather, you can configure a WCF-Custom or WCF-SQL send port to directly invoke the stored procedure. This topic demonstrates how to perform these tasks using the BizTalk Server Administration console.

Note

This topic provides instructions on how to configure a WCF-Custom send port for executing stored procedure that takes a single parameter. You can perform the same steps by configuring a WCF-SQL port. For instructions on configuring WCF-SQL port, see [Configuring a Port Using the WCF-SQL Adapter](#).

How This Topic Demonstrates Invoking Stored Procedures Without an Orchestration

To demonstrate how to execute stored procedures with single parameters without an orchestration, this topic uses the `ADD_LAST_EMP_XML_INFO` stored procedure. This procedure takes an XML value as a parameter and inserts it into the **Address** column of the **Employee** table. You must have the XML value that you will pass to the stored procedure. However, to execute the stored procedure using the adapter, you must send a request message conforming to the schema of the procedure, and containing the XML value for the **Address** field, to the SQL Server. So, you must create that request message by:

- Using the **Template** option in the send port configuration using which you can create a request message using a message template.
- Putting the XML value for the **Address** field into the message.

All these steps are described in detail in this topic. You must perform the following set of tasks:

1. Create a FILE receive port where you will drop the XML file that will be inserted into the **Address** XML field of the **Employee** table. Suppose this port is called **MessageIn** port.
2. Create a WCF-Custom one-way send port that picks the XML file from the FILE receive port, constructs the message using the message template, and sends it to SQL Server to execute the stored procedure.

This part of the topic provides instructions on configuring a WCF-Custom send port with the message template.

Note

Even though the information in this topic demonstrates how to execute a stored procedure with a single XML parameter, you can perform the tasks to perform any operation that takes a single parameter of any data type. The only difference will be in the way you create a message template for a specific operation. You can create a message template by taking the request message you would use to execute the operation using an orchestration and replacing the value of the parameter with the BizTalk message body.

Configuring a WCF-Custom Send Port

Before creating the WCF-Custom send port, make sure you created the FILE receive port, **MessageIn**.

To configure a WCF-Custom send port

1. Start the BizTalk Server Administration console.
2. In the console tree, expand **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the SQL adapter.

4. Right-click **Send Ports**, point to **New**, and then point to **Static One-way Send Port**.
5. In the **Send Port Properties** dialog box, on the **General** tab, type a name for the send port.
6. Configure the port to receive all messages dropped at the FILE receive port, **MessageIn**.
 - a. In the **Send Port Properties** dialog box, from the left pane, click **Filters**.
 - b. In the right pane, under the **Property** column, click the grid, and then select **BTS.ReceivePortName** property.
 - c. For the **Operator** column, select "==".
 - d. For the **Value** column, specify the name of the FILE receive port, **MessageIn**.
7. In the **Send Port Properties** dialog box, on the **General** tab, from the **Type** drop-down list, select **WCF-Custom**, and then click **Configure**.
8. In the **WCF-Custom Transport Properties** dialog box, do the following:
 - a. Click the **General** tab, and in the **Address (URI)** field, specify the connection URI for SQL Server. For more information about the connection URI, see [SQL Server Connection URI](#).
 - b. On the **General** tab, in the **Action** text box, type the action for the operation. See individual topics in [Technical Reference](#) for a list of actions for each operation. For example, the action to invoke the ADD_LAST_EMP_XML_INFO is:

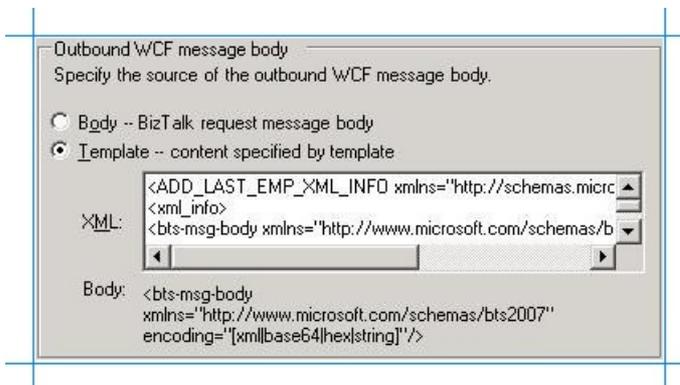
Procedure/dbo/ADD_LAST_EMP_XML_INFO

- c. Click the **Binding** tab, and from the **Binding Type** list, select **sqlBinding**. You can specify the different binding properties exposed by the SQL adapter. For more information about binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).
- d. Click the **Credentials** tab, and then do one of the following:
 - Select the **Do not use Single Sign-On** option, and then specify the user name and password to connect to SQL Server. Note that the user name and password are case-sensitive.

 **Note**

If you want to connect to SQL Server using Windows authentication, specify a blank user name and password.

- Select the **Use Single Sign-On** option, and then specify an affiliate Enterprise Single Sign-on (SSO) application.
- For more information about security with respect to BizTalk Server, see [Security Considerations When Using the Adapter with BizTalk Server](#).
- e. Click the **Messages** tab, and in the **Outbound WCF message body** section, choose the **Template** option.
 - f. In the **XML** text box, specify the template that will be used to construct the WCF message. By doing so, you create a message that conforms to the ADD_LAST_EMP_XML_INFO operation for the WCF-based SQL adapter.



For the ADD_LAST_EMP_XML_INFO stored procedure, you must specify the following template:

```
<ADD_LAST_EMP_XML_INFO xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/dbo">
<xml_info>
<bts-msg-body xmlns="http://www.microsoft.com/schemas/bts2007" encoding="string"/>
</xml_info>
</ADD_LAST_EMP_XML_INFO>
```

Important
The encoding in the message template must always be "string" irrespective of the type of the parameter for the operation that will be invoked using the send port. For example, the ADD_LAST_EMP_XML_INFO takes a parameter of type XML, but the encoding in the message template is string.

Note
You can create this message template by copying the request message for the stored procedure and replacing the value within the <xml_info> tags with the BizTalk message body. You can get the request message for the stored procedure by generating the schema for the procedure using the Consume Adapter Service Add-in, and then generating an instance of the schema to get the request XML.

g. To return to the **Send Port Properties** dialog box, click **OK**.

9. From the **Send handler** list, select **BizTalkServerApplication**.

10. From the **Send pipeline** list, select the pipeline that corresponds to **PassThruTransmit**.

11. Click **OK**.

Starting the Application

To start the BizTalk application, you can start both the FILE receive location and the WCF-Custom send port individually. You must now copy an XML file to the folder mapped to the FILE receive location. The BizTalk application consumes the file, and the XML value is inserted in the Address column of the Employee table. You can verify this by using a SQL Server client and selecting records from the Employee table.

Using a Two-way WCF-Custom Send Port

The instructions in this topic, under the section [Configuring a WCF-Custom Send Port](#), demonstrate how to configure a one-way WCF-Custom send port to execute a stored procedure with a single parameter without using a BizTalk orchestration. However, in such a case, to verify whether the stored procedure is executed successfully you will have to verify in the SQL Server database whether the Address column in the Employee table is updated.

Instead, you can create a two-way WCF-Custom send port that also gets the response from SQL Server if the stored procedure is executed successfully. You must perform a few additional steps if you create a two-way WCF-Custom port. Note that you will still need a FILE receive location, as mentioned in the preceding instructions.

1. Create a two-way WCF-Custom send port, for example, **ExecProcedure**. The steps to configure the send port are similar

to those for the one-way send port. The only difference is that for the two-way port you must also specify a receive pipeline. Make sure you select **PassThruReceive** for the receive pipeline.

2. Create a FILE send port. This port will drop the response message from the SQL Server database to a folder. Using the **Filters** tab of the port properties dialog box, configure the FILE send port to receive all response messages from the WCF-Custom send port.
 - a. In the **Send Port Properties** dialog box, from the left pane, click **Filters**.
 - b. In the right pane, under the **Property** column, click the grid, and then select **BTS.SPName** property.
 - c. For the **Operator** column, select "==".
 - d. For the **Value** column, specify the name of the WCF-Custom send port, **ExecProcedure**.

Start all the three ports. Copy an XML file to the folder mapped to the FILE receive location. Look for the response in the folder mapped to the FILE send port.

See Also

Other Resources

[Developing BizTalk Applications](#)

Executing Stored Procedures Having a FOR XML Clause

This topic was last updated on: June 05, 2009

An SQL SELECT statement can have a FOR XML clause that returns the query result as XML instead of a rowset. You can also have a stored procedure that has a SELECT statement with a FOR XML clause. For more information about the FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

You can use the WCF-based SQL adapter to execute such stored procedures.

Important

The "native" SQL adapter available with BizTalk Server requires stored procedures to have the FOR XML clause as part of the SELECT statement. You can use such stored procedures with the WCF-based SQL adapter without making any changes to the stored procedure definition.

How to Invoke Stored Procedures with FOR XML Clause

When you invoke a stored procedure with FOR XML clause in SQL Server Management Studio or using the SQL adapter available with BizTalk Server, the output is in the form of an xml message. To use these procedures with the WCF-based SQL adapter, you must have the schema for the output message. The WCF-based SQL adapter requires this schema while receiving the response message from SQL Server after executing a stored procedure with FOR XML clause. Note that the request message to invoke this stored procedure will be generated by the adapter itself.

Apart from having the schema for the response message, you must also perform certain tasks to invoke a stored procedure with FOR XML clause using the WCF-based SQL adapter.

1. Generate the schema for the response message for the stored procedure with FOR XML clause. If you already have the response schema generated by the "native" SQL adapter available with BizTalk Server, you can skip this step.
2. Create a BizTalk project and add the generated schema to the project.
3. Generate schema for the stored procedure with FOR XML clause using the WCF-based SQL adapter. This provides the schema for the request message that the adapter sends to SQL Server to invoke the stored procedure.
4. Create messages in the BizTalk project to send and receive messages from SQL Server. The request message must conform to the schema of the request message generated by the adapter. The response message must conform to the schema of the response message obtained either using the "native" SQL adapter or by executing the stored procedure with FOR XML clause in SQL Server Management Studio.
5. Create an orchestration to invoke the stored procedure in the SQL Server database.
6. Build and deploy the BizTalk project.
7. Configure the BizTalk application by creating physical send and receive ports.
8. Start the BizTalk application.

Generating Schema for the Response Message for Stored Procedure

Note

You do not need to perform this step if you have the response schema generated by the SQL adapter available with BizTalk Server.

You can generate the schema for the response message for the stored procedure, provided the SELECT statement in the stored procedure has the `xmlschema` clause with the `for xml` clause. In this topic, we use the `GET_EMP_DETAILS_FOR_XML` stored procedure that retrieves the employee details for a given employee ID. To retrieve the schema by executing the stored procedure, the SELECT statement looks like the following:

```
SELECT [Employee_ID] ,[Name] ,[DOJ] ,[Designation] ,[Job_Description] ,[Photo] ,cast([Rating] as varchar(100)) as Rating ,[Salary] ,[Last_Modified] ,[Status] ,[Address]
FROM [Adapt_Doc].[dbo].[Employee] for xml auto, xmlschema
```

Execute this stored procedure to get the schema for the response message. Note that the response from the stored procedure contains the schema as well as the data from executing the stored procedure. You must copy the schema from the response and save it to a text pad. For this example, you can name this schema as **ResponseSchema.xsd**. You must now create a BizTalk project in Visual Studio and add this schema to the project.

◆ Important

Make sure you remove the `xmlschema` clause after you have executed the stored procedure to generate the schema. If you fail to do this, when you finally execute the stored procedure through BizTalk, you will again generate the schema in the response message. So, to get the response message as xml you must remove the `xmlschema` clause.

To add the schema to a BizTalk project

1. Create a BizTalk project in Visual Studio.
2. Add the response schema you generated for the stored procedure to the BizTalk project. Right-click the BizTalk project in the Solution Explorer, point to **Add**, and then click **Existing Item**. In the Add Existing Item dialog box, navigate to the location where you saved the schema and click **Add**.
3. Open the schema in Visual Studio and make the following changes.
 - a. Add a node to the schema and move the existing root node under this newly added node. Give a name to the root node. For this topic, rename the root node to **Root**.
 - b. The response schema generated for the stored procedure references a `sqltypes.xsd`. You can get the `sqltypes.xsd` schema from <http://go.microsoft.com/fwlink/?LinkId=131087>. Add the `sqltypes.xsd` schema to the BizTalk project.
 - c. In the schema generated for the stored procedure, change the value of `import schemaLocation` to the following.

```
import schemaLocation="sqltypes.xsd"
```

You do this because you have already added the `sqltypes.xsd` schema to your BizTalk project.

- d. Provide a target namespace for the schema. Click the **<Schema>** node, and in the properties pane, specify a namespace in the **Target Namespace** property. For this topic, give the namespace as `http://ForXmlStoredProcs/namespace`.

Generating Schema for the Request Message to Invoke the Stored Procedure

To generate schema for the request message you can use the Consume Adapter Service Add-in from a BizTalk project in Visual Studio. For this topic, generate the schema for the `GET_EMP_DETAILS_FOR_XML` stored procedure. For more information about how to generate the schema using Consume Adapter Service Add-in, see [Retrieving Metadata for SQL Server Operations in Visual Studio](#).

◆ Important

You must generate the schema by selecting the procedure only from the **Procedures** node in Consume Adapter Service Add-in.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration, and link them to schemas that you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**,

and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.

2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

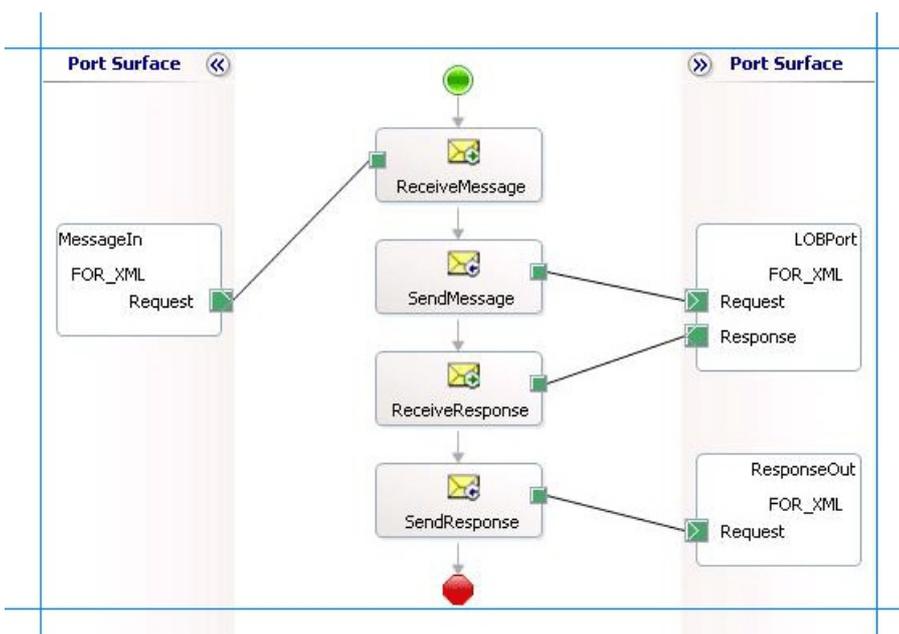
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>ForXMLProcedure.Procedure_dbo.GET_EMP_DETAILS_FOR_XML</i> , where ForXMLProcedure is the name of your BizTalk project. Procedure_dbo is the schema generated for invoking the GET_EMP_DETAILS_FOR_XML procedure.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>ForXMLProcedure.ResponseSchema</i> , where ResponseSchema is the name of the response schema generated by executing the stored procedure as described under Generating Schema for the Response Message for Stored Procedure .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for executing a stored procedure in SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and receive responses, respectively. A sample orchestration for invoking a procedure resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.FOR_XML.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.FOR_XML.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.FOR_XML.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.FOR_XML.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to SQL Server database. For instructions on how to create a send port, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

You must also specify the action in the send port. For procedures that contain the FOR XML clause, you must set the action in the following format.

```
XmlProcedure/<schema_name>/<procedure_name>
```

So, for this topic where we are executing the GET_EMP_DETAILS_FOR_XML procedure, the action will be:

```
XmlProcedure/dbo/GET_EMP_DETAILS_FOR_XML
```

For more information about setting action, see [Specifying SOAP Action](#).

You must also set the following binding properties when executing a stored procedure with the FOR XML clause.

Binding property name	Set this to
XmlStoredProcedureRootNodeName	Specify the name of the root node that you added to the response schema you generated for the stored procedure, as described under Generating Schema for the Response Message for Stored Procedure . For this topic, set this to Root .
XmlStoredProcedureRootNamespace	Specify the target namespace for the response schema you generated for the stored procedure, as described under Generating Schema for the Response Message for Stored Procedure . For this topic, set this to <code>http://ForXmlStoredProcs/namespace</code> .

For more information about specifying values for binding properties, see [Specifying Binding Properties](#).

Note
Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see Configuring a Physical Port Binding Using a Port Binding File .

Starting the Application

You must start the BizTalk application for invoking procedures in SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the request schema for the procedure you generated using the Consume Adapter Service Add-in. For example, the request message to invoke the GET_EMP_DETAILS_FOR XML is:

```
<GET_EMP_DETAILS_FOR_XML xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/dbo">
  <emp_id>10765</emp_id>
</GET_EMP_DETAILS_FOR_XML>
```

See [Message Schemas for Procedures and Functions](#) for more information about the request message schema for invoking procedures in SQL Server database using the SQL adapter.

The orchestration consumes the message and sends it to SQL Server database. The response from SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<Root xmlns="http://ForXmlStoredProcs/namespace">
  <Adapt_Doc.dbo.Employee Employee_ID="10765" Name="John" Designation="asdfaf" Salary="3434.00" Last_Modified="AAAAAAAAANso=" Status="0" xmlns="" />
</Root>
```

Notice that the response is received in the same schema as generated by executing the stored procedure. Also note that the root node and the namespace is the same you specified as values for **XmlStoredProcedureRootNodeName** and **XmlStoredProcedureRootNodeNamespace** binding properties respectively.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing Composite Operations on SQL Server by Using BizTalk Server

The SQL adapter enables adapter clients to perform composite operations on the SQL Server database. A composite operation can include:

- Insert, Update, and Delete operations. A Select operation is not supported as part of a composite operation.
- Stored procedures executed as operations.

A single composite operation can have any number of these operations, in any order. For example, you can have two Insert operations followed by a Delete operation, and finally a stored procedure execution. Also, you can have different operations targeting different database tables or views. For more information about how the adapter supports composite operations, see [Support for Composite Operations](#). For information about the structure of the SOAP message for composite operations, see [Message Schemas for Composite Operations](#).

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

How to Perform Composite Operations on SQL Server

Performing an operation on SQL Server using the SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To perform composite operations on the SQL Server database, these tasks are:

1. Create a BizTalk project, and generate schema for all the operations you want to invoke.
2. Manually create a schema file that includes references to all the schemas you generated in the previous step.
3. Create messages in the BizTalk project for sending and receiving messages from the SQL Server database. These messages must conform to the request and response schema you created in the previous step.
4. Create an orchestration to invoke the composite operation on the SQL Server database.
5. Build and deploy the BizTalk project.
6. Configure the BizTalk application by creating physical send and receive ports.
7. Start the BizTalk application.

This topic provides instructions on how to perform these tasks.

Sample Based on This Topic

A sample, CompositeOperations, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

In this topic, to demonstrate how to perform composite operations, the following tasks will be performed in the order specified:

- Insert record into the EMPLOYEE table.
- Retrieve all the columns for the last inserted record by invoking the GET_LAST_EMP_DATA stored procedure.

- Delete the record from the EMPLOYEE table.

Run the scripts provided with the samples to create the EMPLOYEE table. For more information about the samples, see [Samples](#).

You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema for these operations. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schemas.

Creating a Composite Schema Definition

You must now create a composite schema that references the schemas you created for the individual operations. Perform the following steps to create a composite schema definition.

To add a composite schema definition

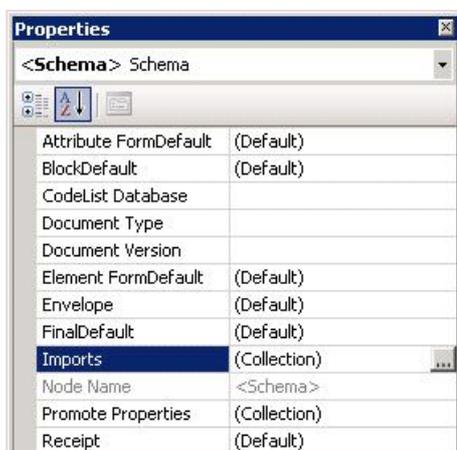
1. Add a schema file to the BizTalk project. Right-click the project name, point to **Add**, and then click **New Item**. In the **Add New Item** dialog box, from the **Categories** box, click **Schema Files**. From the **Templates** box, click **Schema**. Specify a name for the schema file, and then click **OK**.

For this example, specify the schema file name as **CompositeSchema.xsd**.

2. Add references to the schema generated for the different operations that you want to perform. In this example, the different schemas generated for operations are:
 - TableOperation.dbo.Employee.xsd, for Insert and Delete operations.
 - Procedure.dbo.xsd, for the GET_LAST_EMP_DATA stored procedure.

To add references:

- a. Right-click the root **<Schema>** node in the CompositeSchema.xsd, and click **Properties**.
- b. In the **Property** box, click the ellipsis button (...) against the **Imports** property.



- c. In the **Imports** dialog box, from the **Import new schema as** list, select **XSD Import**, and then click **Add**.
- d. In the **BizTalk Type Picker** dialog box, expand the BizTalk project name node, expand **Schemas**, and then select the schema you want to import. For this example, select **<BizTalk_project_name>.TableOperation_dbo_Employee**. Click **OK**.

Repeat this step to import **<BizTalk_project_name>.Procedure_dbo** too.

- e. In the **Imports** dialog box, click **OK**.

3. Add two child nodes to the root schema node. One child node corresponds to the request schema for performing the composite operation. The other child node corresponds to the response schema. The node that corresponds to the request schema can have any name. The node that corresponds to the response schema must be called **<request_schema_node>Response**. For this example, we will call the request schema node as **Request**. So, the response

schema node is called **RequestResponse**.

Note

By default, a **Root** node is also added to a new schema file. You can rename the **Root** node to **Request**. To rename a node, right-click the node name and click **Rename**.

To add a node under the **<Schema>** node:

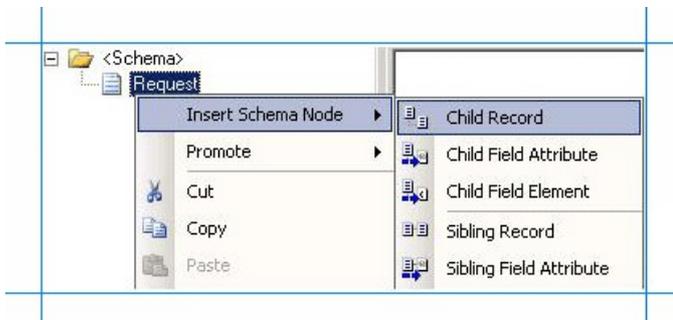
- a. Right-click the **<Schema>** node, point to **Insert Schema Node**, and click **Child Record**.
 - b. Rename the new node to **RequestResponse**.
4. Add child nodes under the **Request** node that correspond to the request schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:
- Insert and Delete operations on the EMPLOYEE table.
 - GET_LAST_EMP_DATA stored procedure.

Important

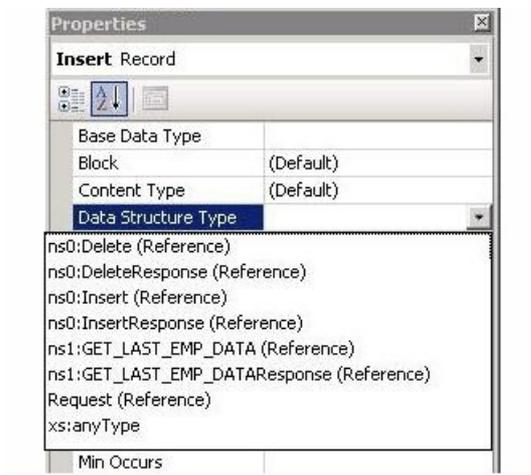
You must add the nodes in the same order in which you want to perform the operations. For example, if you want to insert a record, then execute a stored procedure, and then delete a record you must first add a node for the Insert operation, followed by a node for the stored procedure, and finally a node for the Delete operation.

To add child nodes to the **Request** node:

- a. Right-click the **Request** node, point to **Insert Schema Node**, and then click **Child Record**.



- b. Rename the record to correspond to a request schema for an operation that you perform as part of the composite operation. For example, rename the node to "Insert".
- c. Map the **Insert** node to the request schema for the Insert operation on the EMPLOYEE table. To do so, right-click the **Insert** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **Insert (Reference)**.



- d. Repeat these steps to add nodes for the request schemas for GET_LAST_EMP_DATA stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following table.

Node name	Mapped to schema
GET_LAST_EMP_DATA	GET_LAST_EMP_DATA (Reference)
Delete	Delete (Reference)

5. Add child nodes under the **RequestResponse** node that correspond to the response schema for each operation that you will perform as part of the composite operation. For this example, you must add child nodes corresponding to the following:

- Insert and Delete operations on the EMPLOYEE table.
- GET_LAST_EMP_DATA stored procedure.

Important
 You must add the child nodes in the same order as the child nodes under the **Request** node.

To add child nodes to the **RequestResponse** node:

- Right-click the **RequestResponse** node, point to **Insert Schema Node**, and click **Child Record**.
- Rename the record to correspond to a response schema for an operation that you perform as part of the composite operation. For example, rename the node to "InsertResponse".
- Map the **InsertResponse** node to the response schema for the Insert operation on the EMPLOYEE table. To do so, right-click the **InsertResponse** node, and click **Properties**. In the **Properties** box, from the **Data Structure Type** list, select **InsertResponse (Reference)**.
- Repeat these steps to add nodes for the response schemas for the GET_LAST_EMP_DATA stored procedure and the Delete operation. Specify the node names and map them to the corresponding schema as mentioned in the following table.

Node name	Mapped to schema
GET_LAST_EMP_DATAResponse	GET_LAST_EMP_DATAResponse (Reference)
DeleteResponse	DeleteResponse (Reference)

6. Save the **CompositeSchema.xsd** file.

Defining Messages and Message Types

The composite schema that you created in the last step describes the “types” required for the messages in an orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schema you created in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

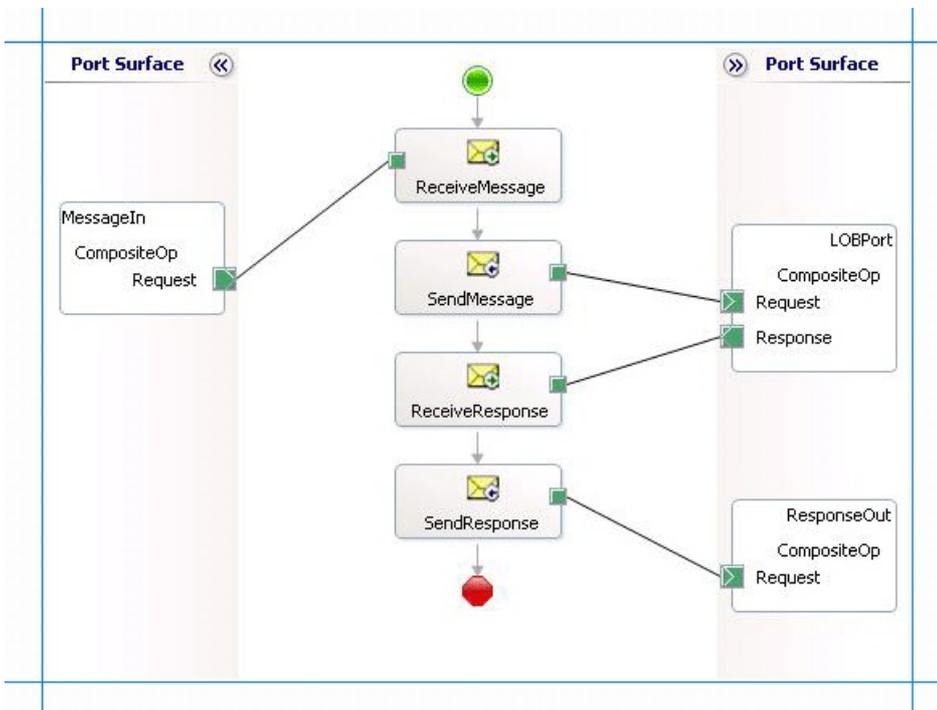
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>CompositeOp.CompositeSchema.Request</i> , where CompositeOp is the name of your BizTalk project. CompositeSchema is the schema you created manually for the composite operations.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>CompositeOp.CompositeSchema.RequestResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing composite operations on SQL Server. In this orchestration, you drop a request message at a defined receive location. The request message must conform to the composite schema you created earlier. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and receive responses, respectively. A basic orchestration for performing composite operations resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> Set Identifier to <i>MessageIn</i> Set Type to <i>MessageInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>

LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.CompositeOp.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.CompositeOp.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.CompositeOp.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.CompositeOp.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console.

For this orchestration you must:

- Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
- Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. Because the operations that are being as part of the composite operation are executed in a single transaction, make sure the **UseAmbientTransaction** binding property is set to **True**.

You must also specify the action in the send port. The action for a composite operation is "**CompositeOperation**". For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). For more information about how to specify actions for ports, see [Specifying SOAP Action](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#). If you import this binding file, the action on the WCF-Custom or WCF-SQL send port is set to a dynamic action involving all the operations you selected in the Consume Adapter Service Add-in while generating the schema. For a composite operation, you must replace the dynamic action with "CompositeOperation".

Starting the Application

You must start the BizTalk application for performing composite operations on the SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to the SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the composite operations you created earlier. For example, a request message that inserts a record in the EMPLOYEE table, invokes the GET_LAST_EMP_DATA stored procedure, and deletes a record from the EMPLOYEE table is:

```
<Request xmlns="http://CompositeTest.CompositeSchema">
  <Insert xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
    <Rows>
      <Employee xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/Tables/dbo">
        <Name>John</Name>
        <Designation>Manager</Designation>
        <Salary>100000</Salary>
      </Employee>
    </Rows>
  </Insert>
  <GET_LAST_EMP_DATA xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/dbo" />
  <Delete xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
```

```

<Rows>
  <Employee xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/Tables/dbo">
    <Name>John</Name>
  </Employee>
</Rows>
</Delete>
</Request>

```

See [Message Schemas for Composite Operations](#) for more information about the request message schema for performing composite operations on the SQL Server database using the SQL adapter.

The orchestration consumes the message and sends it to the SQL Server database. The response from the SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from the SQL Server database for the preceding request message is:

```

<?xml version="1.0" encoding="utf-8" ?>
<RequestResponse xmlns="http://CompositeTest.CompositeSchema">
  <InsertResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
    <InsertResult>
      <long xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">10080</long>
    </InsertResult>
  </InsertResponse>
  <GET_LAST_EMP_DATAResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/dbo">
    <GET_LAST_EMP_DATAResult>
      <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data">
        <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
          <xs:element msdata:IsDataSet="true" name="NewDataSet">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element minOccurs="0" name="Employee_ID" type="xs:int" />
                      <xs:element minOccurs="0" name="Name" type="xs:string" />
                      <xs:element minOccurs="0" name="DOJ" type="xs:dateTime" />
                      <xs:element minOccurs="0" name="Designation" type="xs:string" />
                      <xs:element minOccurs="0" name="Job_Description" type="xs:string" />
                      <xs:element minOccurs="0" name="Photo" type="xs:base64Binary" />
                      <xs:element minOccurs="0" name="Rating" type="xs:string" />
                      <xs:element minOccurs="0" name="Salary" type="xs:decimal" />
                      <xs:element minOccurs="0" name="Last_Modified" type="xs:base64Binary" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:schema>
        <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
          <NewDataSet xmlns="">
            <NewTable>
              <Employee_ID>10080</Employee_ID>
              <Name>John</Name>
              <Designation>Manager</Designation>
              <Salary>100000.00</Salary>
              <Last_Modified>AAAAAAAAAF40=</Last_Modified>
            </NewTable>
          </NewDataSet>
        </diffgr:diffgram>
      </DataSet>
    </GET_LAST_EMP_DATAResult>
    <ReturnValue>0</ReturnValue>
  </GET_LAST_EMP_DATAResponse>
  <DeleteResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">

```

```
<DeleteResult>1</DeleteResult>
</DeleteResponse>
</RequestResponse>
```

The preceding response contains multiple result sets corresponding the different operations performed as part of the composite operation. For example, the `InsertResult` element contains 10080, which is the unique identifier for the newly added record.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Scalar Functions in SQL Server by Using BizTalk Server

You can use the SQL adapter with BizTalk Server to invoke scalar functions in SQL Server. The adapter exposes the scalar functions as operations that can be invoked directly on SQL Server. For more information about how the adapter supports scalar functions, see [Executing Scalar Functions in SQL Server](#). For information about the structure of the SOAP message for invoking scalar functions, see [Message Schemas for Procedures and Functions](#).

Prerequisites

- You must have completed the steps in [Before You Develop BizTalk Applications](#).
- You must have configured MSDTC on computers running the SQL adapter and SQL Server. See [Before Developing Applications Using the SQL Adapter](#).

How to Invoke Scalar Functions on SQL Server Database

Performing an operation on SQL Server database by using SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke scalar functions in SQL Server, these tasks are:

1. Create a BizTalk project, and generate schema for the scalar function you want to invoke in SQL Server.
2. Create messages in the BizTalk project for sending and receiving messages from SQL Server.
3. Create an orchestration to invoke the operation on SQL Server.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke scalar functions in SQL Server using the SQL adapter with BizTalk Server. To demonstrate this operation, in this topic you execute the GET_EMP_ID function. This function takes an employee's designation as a parameter and returns the ID for that employee from the EMPLOYEE table. The table and the function are created by running the scripts provided with the samples. For more information about the script, see [Samples](#).

To demonstrate how to invoke scalar functions, schema is generated for the GET_EMP_ID scalar function. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schemas.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.

5. In the **Properties** pane for the **Message_1**, do the following:

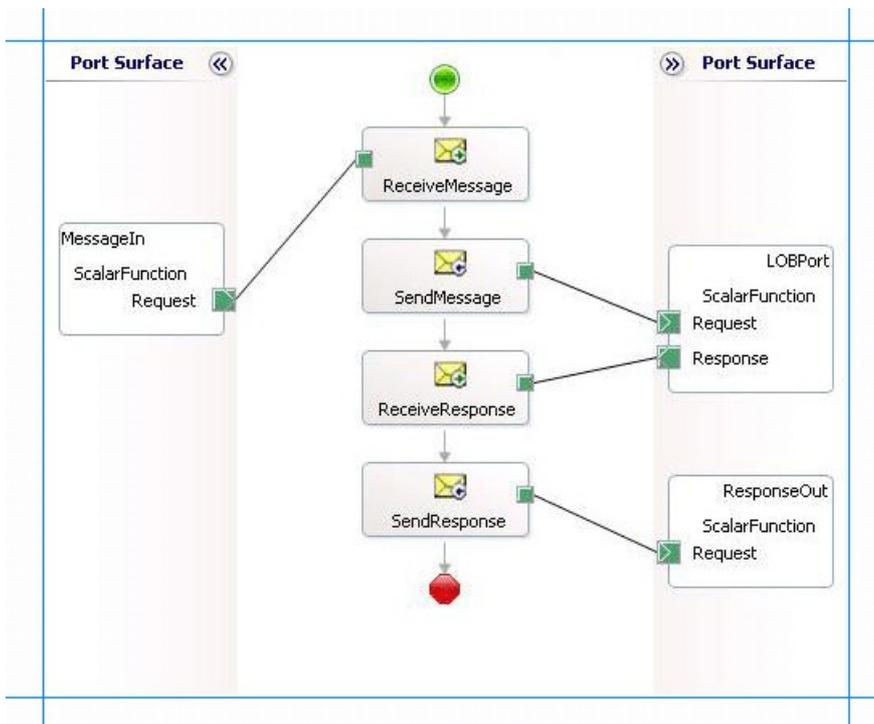
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>ScalarFunction.ScalarFunction_dbo.GET_EMP_ID</i> , where <i>ScalarFunction</i> is the name of your BizTalk project. <i>ScalarFunction_dbo</i> is the schema generated for the GET_EMP_ID function.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>ScalarFunction.ScalarFunction_dbo.GET_EMP_IDResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and to receive responses, respectively. A sample orchestration for invoking a scalar function resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>

SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.ScalarFunction.Request</i>

SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.ScalarFunction.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.ScalarFunction.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.ScalarFunction.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking scalar functions in SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.

- The WCF-Custom or WCF-SQL send port to send messages to SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the GET_EMP_ID function you generated earlier. For example, the request message to invoke the GET_EMP_ID function is:

```
<GET_EMP_ID xmlns="http://schemas.microsoft.com/Sql/2008/05/ScalarFunctions/dbo">
  <emp_desig>Manager</emp_desig>
</GET_EMP_ID>
```

This request message invokes the GET_EMP_ID function to retrieve the ID for employees with the designation "Manager". See [Message Schemas for Procedures and Functions](#) for more information about the request message schema for invoking scalar functions in SQL Server using the SQL adapter.

The orchestration consumes the message and sends it to SQL Server database. The response from SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8"?>
<GET_EMP_IDResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/ScalarFunctions/dbo">
  <GET_EMP_IDResult>10072</GET_EMP_IDResult>
</GET_EMP_IDResponse>
```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Invoking Table-Valued Functions in SQL Server by Using BizTalk Server

You can use the SQL adapter with BizTalk Server to invoke table-valued functions in SQL Server. The adapter exposes the table-valued functions as operations that can be invoked directly on SQL Server. For more information about how the adapter supports table-valued functions, see [Executing Table-Valued Functions in SQL Server](#). For information about the structure of the SOAP message for invoking table-valued functions, see [Message Schemas for Procedures and Functions](#).

Prerequisites

- You must have completed the steps in [Before You Develop BizTalk Applications](#).
- You must have configured MSDTC on computers running the SQL adapter and SQL Server. See [Before Developing Applications Using the SQL Adapter](#).

How to Invoke Table-valued Functions on a SQL Server Database

Performing an operation on a SQL Server database by using SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke table-valued functions in SQL Server, these tasks are:

1. Create a BizTalk project, and generate schema for the table-valued function you want to invoke in SQL Server.
2. Create messages in the BizTalk project for sending and receiving messages from SQL Server.
3. Create an orchestration to invoke the operation on SQL Server.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke table-valued functions in SQL Server using the SQL adapter with BizTalk Server. To demonstrate this operation, in this topic you execute the TVF_EMPLOYEE function. This function takes an employee's designation as a parameter and returns all the employee records with that designation, as a table type. The Employee table and the function are created by running the scripts provided with the samples. For more information about the script, see [Samples](#).

To demonstrate how to invoke table-valued functions, schema is generated for the TVF_EMPLOYEE table-valued function. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schemas.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.
2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.

- In Orchestration View, right-click **Messages**, and then click **New Message**.
- Right-click the newly created message, and then select **Properties Window**.
- In the **Properties** pane for the **Message_1**, do the following:

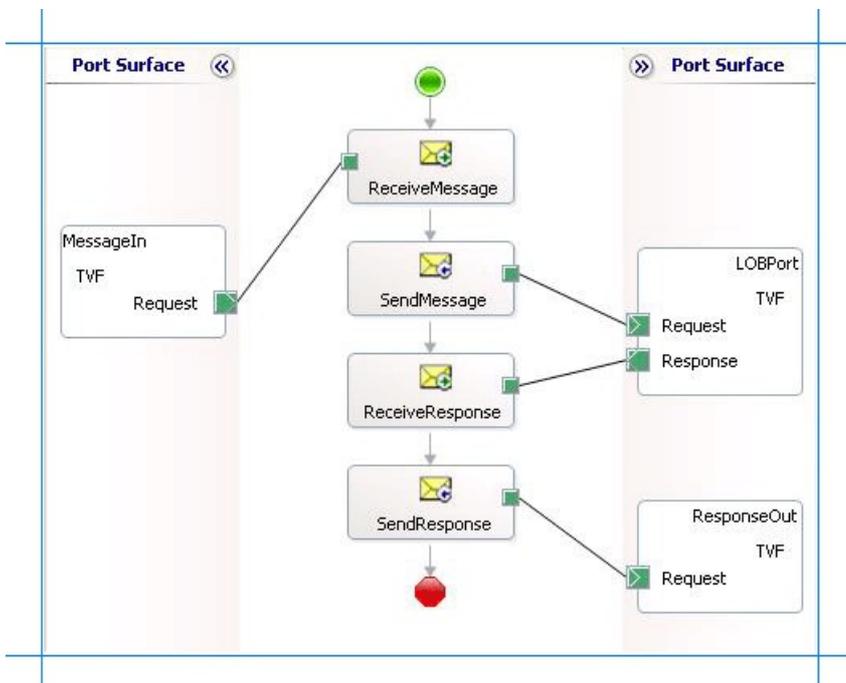
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>TableFunctions.TableValuedFunction_dbo.TVF_EMPLOYEE</i> , where TableFunctions is the name of your BizTalk project. TableValuedFunction_dbo is the schema generated for the TVF_EMPLOYEE function.

- Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>TableFunctions.TableValuedFunction_dbo.TVF_EMPLOYEEResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and to receive responses, respectively. A sample orchestration for invoking a table-valued function resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>

SendMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveResponse</i> Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> Set Identifier to <i>MessageIn</i> Set Type to <i>MessageInType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> Set Identifier to <i>LOBPort</i> Set Type to <i>LOBPortType</i> Set Communication Pattern to <i>Request-Response</i> Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> Set Identifier to <i>ResponseOut</i> Set Type to <i>ResponseOutType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> Set Message to <i>Request</i> Set Operation to <i>MessageIn.TVF.Request</i>
SendMessage	<ul style="list-style-type: none"> Set Message to <i>Request</i> Set Operation to <i>LOBPort.TVF.Request</i>

ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.TVF.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.TVF.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking table-valued functions in the SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to the SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the TVF_EMPLOYEE table-valued function you generated earlier. For example, the request message to invoke the TVF_EMPLOYEE function is:

```
<TVF_EMPLOYEE xmlns="http://schemas.microsoft.com/Sql/2008/05/TableValuedFunctions/dbo">
  <emp_desig>Tester</emp_desig>
</TVF_EMPLOYEE>
```

This request message invokes the TVF_EMPLOYEE function to retrieve employee records having the designation as "Tester". See [Message Schemas for Procedures and Functions](#) for more information about the request message schema for invoking table-valued functions in SQL Server using the SQL adapter.

Note

If you do not specify a value for the parameter the adapter internally executes the function using the DEFAULT keyword, which means that the adapter executes the function by passing the default value for the parameter, if available as part of the function definition.

The orchestration consumes the message and sends it to the SQL Server database. The response from the SQL Server database is saved at the other FILE location defined as part of the orchestration. For example, the response from the SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<TVF_EMPLOYEEResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableValuedFunctions/dbo">
  <TVF_EMPLOYEEResult>
    <TVF_EMPLOYEE xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/TableFunctionReturnTables/dbo">
      <Employee_ID>10499</Employee_ID>
      <Name>John</Name>
      <Designation>Tester</Designation>
      <Salary>999999.00</Salary>
      <Last_Modified>AAAAAAAAAJBM=</Last_Modified>
    </TVF_EMPLOYEE>
    <TVF_EMPLOYEE xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/TableFunctionReturnTables/dbo">
      .....
      .....
    </TVF_EMPLOYEE>
    .....
    .....
  </TVF_EMPLOYEEResult>
</TVF_EMPLOYEEResponse>
```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using BizTalk Server

The SQL adapter exposes generic SQL Server operations such as **ExecuteNonQuery**, **ExecuteReader**, and **ExecuteScalar**. You can use these operations to execute any SQL statement on a SQL Server database. These operations differ based on the kind of response you get for the SQL statement. For more information about how the adapter supports these operations, see [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

This topic demonstrates how to perform an **ExecuteReader** operation using the SQL adapter with BizTalk Server. You can follow the same set of procedures described in this topic to perform **ExecuteNonQuery** and **ExecuteScalar** operations.

Prerequisites

- You must have completed the steps in [Before You Develop BizTalk Applications](#).
- You must have configured MSDTC on computers running the SQL adapter and SQL Server. See [Before Developing Applications Using the SQL Adapter](#).

How to Invoke the ExecuteReader operation on a SQL Server Database

Performing an operation on a SQL Server database by using SQL adapter with BizTalk Server involves procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To invoke the **ExecuteReader** operation in SQL Server, these tasks are:

1. Create a BizTalk project, and generate schema for the **ExecuteReader** operation.
2. Create messages in the BizTalk project for sending and receiving messages from SQL Server.
3. Create an orchestration to invoke the operation on SQL Server.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.
6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

This topic demonstrates how to invoke **ExecuteReader** operation in SQL Server using the SQL adapter with BizTalk Server. The **ExecuteReader** operation takes any SQL statement as a parameter and returns the result set for the operation as an array of DataSet. This topic shows how to execute the ADD_EMP_DETAILS stored procedure using the **ExecuteReader** operation. This stored procedure adds a record to the Employee table and returns the ID for the newly added employee. The tables and stored procedures used in the topic are created by running the scripts provided with the samples. For more information about the script, see [Samples](#).

To demonstrate how to invoke **ExecuteReader** operation, schema is generated for the **ExecuteReader** operation. You must create a BizTalk project and use the Consume Adapter Service Add-in to generate the schema. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate schemas.

Defining Messages and Message Types

The schema that you generated earlier describes the “types” required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. You must now create messages for the orchestration and link them to schemas you generated in the previous step.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From Solution Explorer, right-click the BizTalk project name, point to **Add**,

and then click **New Item**. Type a name for the BizTalk orchestration, and then click **Add**.

2. Open the Orchestration View window of the BizTalk project, if it is not already open. To do so, click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In Orchestration View, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for the **Message_1**, do the following:

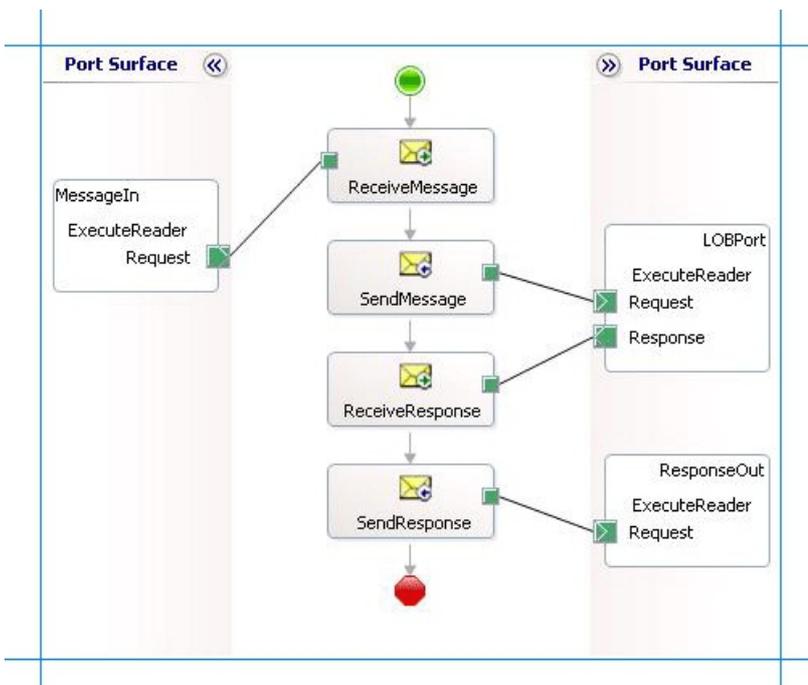
Use this	To do this
Identifier	Type Request
Message Type	From the drop-down list, expand Schemas , and then select <i>Execute_Reader.GenericOperation.ExecuteReader</i> , where <i>Execute_Reader</i> is the name of your BizTalk project. <i>GenericOperation</i> is the schema generated for the ExecuteReader operation.

6. Repeat step 2 to create a new message. In the **Properties** pane for the new message, do the following:

Use this	To do this
Identifier	Type Response
Message Type	From the drop-down list, expand Schemas , and then select <i>Execute_Reader.GenericOperation.ExecuteReaderResponse</i> .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for performing an operation on SQL Server. In this orchestration, you drop a request message at a defined receive location. The SQL adapter consumes this message and passes it on to SQL Server. The response from SQL Server is saved to another location. You must include Send and Receive shapes to send messages to SQL Server and to receive responses, respectively. A sample orchestration for invoking an **ExecuteReader** operation resembles the following:



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SendMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SendMessage</i>
ReceiveResponse	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveResponse</i> • Set Activate to <i>False</i>
SendResponse	Send	<ul style="list-style-type: none"> • Set Name to <i>SendResponse</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
MessageIn	<ul style="list-style-type: none"> • Set Identifier to <i>MessageIn</i> • Set Type to <i>MessageInType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
LOBPort	<ul style="list-style-type: none"> • Set Identifier to <i>LOBPort</i> • Set Type to <i>LOBPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>
ResponseOut	<ul style="list-style-type: none"> • Set Identifier to <i>ResponseOut</i> • Set Type to <i>ResponseOutType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes, and Connect Them to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
-------	------------

ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>MessageIn.ExecuteReader.Request</i>
SendMessage	<ul style="list-style-type: none"> • Set Message to <i>Request</i> • Set Operation to <i>LOBPort.ExecuteReader.Request</i>
ReceiveResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>LOBPort.ExecuteReader.Response</i>
SendResponse	<ul style="list-style-type: none"> • Set Message to <i>Response</i> • Set Operation to <i>ResponseOut.ExecuteReader.Request</i>

After you have specified these properties, the message shapes and ports are connected, and your orchestration is complete.

You must now build the BizTalk solution and deploy it to BizTalk Server. For more information, see <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the Orchestrations pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where you will drop a request message. The BizTalk orchestration will consume the request message and send it to the SQL Server database.
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the response message containing the response from the SQL Server database.
 - Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. You must also specify the action in the send port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#).

Note

Generating the schema using the Consume Adapter Service BizTalk Project Add-in also creates a binding file that contains information about the ports and the actions to be set for those ports. You can import this binding file from the BizTalk Server Administration console to create send ports (for outbound calls) or receive ports (for inbound calls). For more information, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Starting the Application

You must start the BizTalk application for invoking the **ExecuteReader** operation in the SQL Server database. For instructions on starting a BizTalk application, see <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The FILE receive port to receive request messages for the orchestration is running.
- The FILE send port to receive the response messages from the orchestration is running.
- The WCF-Custom or WCF-SQL send port to send messages to the SQL Server database is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, you must drop a request message to the FILE receive location. The schema for the request message must conform to the schema for the **ExecuteReader** operation you generated earlier. For example, the request message to invoke the ADD_EMP_DETAILS using an **ExecuteReader** operation is:

```
<ExecuteReader xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/">
  <Query>EXEC ADD_EMP_DETAILS Tom,Manager,100000</Query>
</ExecuteReader>
```

See [Message Schemas for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#) for more information about the request message schema for invoking an **ExecuteReader** operation using the SQL adapter.

Note

Within the `<Query>` tag, you can specify multiple SQL statements separated by a semi-colon.

The orchestration consumes the message and sends it to the SQL Server database. The response from the SQL Server database is saved at the other FILE location defined as part of the orchestration. The response for the **ExecuteReader** operation contains a result set as a DataSet array. For example, the response from the SQL Server database for the preceding request message is:

```
<?xml version="1.0" encoding="utf-8" ?>
<ExecuteReaderResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/">
  <ExecuteReaderResult>
    <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data">
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" name="Employee_ID" type="xs:int" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
      <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
        <NewDataSet xmlns="">
          <NewTable>
            <Employee_ID>10767</Employee_ID>
          </NewTable>
        </NewDataSet>
      </diffgr:diffgram>
    </DataSet>
  </ExecuteReaderResult>
</ExecuteReaderResponse>
```

In the response, 10767 is the ID of the newly created employee.

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create items such as the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Developing BizTalk Applications](#)

Polling SQL Server by Using the SQL Adapter with BizTalk Server

You can configure the SQL adapter to receive polling-based data-changed messages from SQL Server. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set. Based on the type of polling message received, the adapter exposes three different ways of polling:

- **Polling.** This operation returns a data set as part of the polling message. At design time, the schema of the database object being polled is not available. Instead, the schema is available as part of the polling message during run time.
- **TypedPolling.** This operation returns a strongly-typed polling message. At design time, the schema of the database object is also available. You must use this operation for polling if you want to map certain elements from the polling message to another schema, which could be for another operation.
- **XmlPolling.** This operation returns the polling message as an XML message. You must use this operation if you want to use SELECT statements or stored procedures that use the FOR XML clause to return data as XML messages. For more information about the FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

For more information about these polling operations, see [Support for Polling](#).

Note

The SQL adapter enables adapter clients to have a single BizTalk application with more than one Polling or TypedPolling operations for the same database or table. To support such a scenario, the adapter includes a unique ID— **InboundID**—in the connection URI. This ID, when added to the connection URI, makes it unique, thereby enabling multiple polling operations in a single BizTalk application.

The topics in this section provide instructions on how to use Polling, TypedPolling, and XmlPolling in a BizTalk application. This section also provides information about how to use the **InboundID** connection property.

In This Section

- [Receiving Polling-based Data-changed Messages from SQL Server by Using BizTalk Server](#)
- [Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using BizTalk Server](#)
- [Receiving Polling Messages Using SELECT Statements with FOR XML Clause](#)
- [Receiving Polling Messages Across Multiple Receive Ports](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Receiving Polling-based Data-changed Messages from SQL Server by Using BizTalk Server

You can configure the SQL adapter to receive periodic data-change messages for SQL Server tables or views. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set.

For more information on how the adapter supports polling, see [Support for Polling](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling and TypedPolling Operations](#).

Note

This topic demonstrates how to use the **Polling** inbound operation to use polling messages. The message for the Polling operation is not strongly-typed and schema of the object being polled is retrieved along with the message at run-time. If you want to get strongly-typed polling message, you must use the **TypedPolling** operation. You must also use the **TypedPolling** operation to have multiple polling operations in a single BizTalk application. For instructions on how to perform **TypedPolling** operation, see [Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using BizTalk Server](#).

Important

If you want to have more than one polling operation in a single BizTalk application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. With a unique connection URI, you can create multiple receive ports that poll the same database, or even the same table in a database. For more information, see [Receiving Polling Messages Across Multiple Receive Ports](#).

How This Topic Demonstrates Polling

In this topic, to demonstrate how the SQL adapter supports receiving data change messages, create a BizTalk project and generate schema for the **Polling** operation. If you want to specify the polling related binding properties at design time, specify the **PolledDataAvailableStatement** as:

```
SELECT COUNT(*) FROM Employee
```

The **PolledDataAvailableStatement** must return a result set with the first cell containing a positive value. If the first cell does not contain a positive value, the adapter does not execute the polling statement.

As part of the polling statement, perform the following operations:

- Select all the rows from the Employee table.
- Execute a stored procedure (MOVE_EMP_DATA) to move all the records from the Employee table to an EmployeeHistory table.
- Execute a stored procedure (ADD_EMP_DETAILS) to add a new record to the Employee table. This procedure takes the employee name, designation, and salary as parameters.

To perform these operations, you must specify the following for the **PollingStatement** binding property:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000
```

After the polling statement is executed, all the records from the Employee table are selected and the message from SQL Server is dropped to a receive location. Once the MOVE_EMP_DATA stored procedure is executed by the adapter, all the records are moved to the EmployeeHistory table. Then, the ADD_EMP_DETAILS stored procedure is executed to add a new record to the Employee table. The next polling execution will only return a single record. This cycle continues until you disable the receive port that polls SQL Server.

Configuring a Polling Query with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure the adapter to receive data-

change messages. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

Note
 You may choose to specify these binding properties when generating the schema for the **Polling** operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-SQL receive port with the binding properties already set. For more information about creating a port using the binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling , TypedPolling , or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. The SQL statement must return a result set consisting of rows and columns. Only if a row is available, the SQL statement specified for the PollingStatement binding property will be executed.
PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter waits for the remaining time in the interval.
PollingStatement	Specifies the SQL statement to poll the SQL Server database table. You can specify a simple SELECT statement or a stored procedure for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property. You can specify any number of SQL statements separated by a semicolon.
PollWhenDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false .

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to poll SQL Server, read further.

How to Receive Data-change Messages from the SQL Server Database

Performing an operation on the SQL Server database using SQL adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive data-change messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Polling** operation. Optionally, you can specify values for the **PolledDataAvailableStatement** and **PollingStatement** binding properties.
2. Create a message in the BizTalk project for receiving messages from the SQL Server database.
3. Create an orchestration to receive messages from the SQL Server database and to save them to a folder.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

◆ Important

For inbound polling scenarios you must always configure a one-way WCF-Custom or WCF-SQL receive port. Two-way WCF-Custom or WCF-SQL receive ports are not supported for inbound operations.

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Polling** operation. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **PolledDataAvailableStatement** and **PollingStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

For instructions on how to specify binding properties, see [Specifying Binding Properties](#).

2. Select the contract type as **Service (Inbound operation)**.
3. Generate schema for the **Polling** operation.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from the SQL Server database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type Receive .
Message Type	From the drop-down list, expand Schemas , and select <i>PollingQuery.Polling</i> , where <i>PollingQuery</i> is the name of your BizTalk project. <i>Polling</i> is the schema generated for the Polling operation.

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from the SQL Server database. In this orchestration, the adapter receives the response of the select statement specified for the **PollingStatement** binding property. The response for the SELECT statement is saved to a FILE location. A typical orchestration

for polling a SQL Server database would contain:

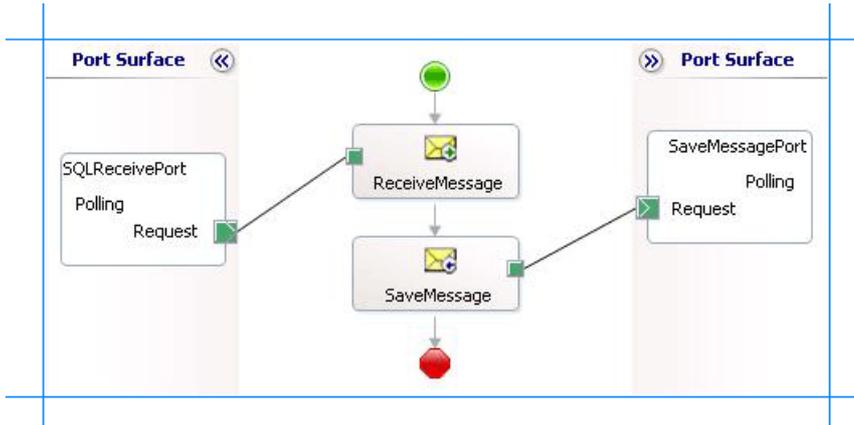
- Receive and Send shapes to receive messages from SQL Server and send to a FILE port, respectively.
- A one-way receive port to receive messages from SQL Server.

Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from a SQL Server database to a folder.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SaveMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveMessage</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
SQLReceivePort	<ul style="list-style-type: none"> • Set Identifier to <i>SQLReceivePort</i> • Set Type to <i>SQLReceivePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>

SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>
-----------------	---

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SQLReceivePort.Polling.Request</i>
SaveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SaveMessagePort.Polling.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the messages from the SQL Server database. These messages will be in response to the polling statement that you specify for the receive port.
 - Define a physical WCF-Custom or WCF-SQL one-way receive port. This port polls the SQL Server database with the polling statement you specify for the port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Make sure you specify the following binding properties for the receive port.

◆ Important

You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-SQL receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
------------------	-------

InboundOperationType	Make sure you set this to Polling .
PolledDataAvailableStatement	Make sure you specify a SQL statement. For this topic, specify: <pre>SELECT COUNT(*) FROM Employee</pre>
PollingStatement	Make sure you specify the polling statement. For this topic, specify: <pre>SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000</pre>

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note
We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the SQL adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-SQL receive port. For instruction on how to add the service behavior, see Configure Transaction Isolation Level and Transaction Timeout .

Starting the Application

You must start the BizTalk application for receiving messages from the SQL Server database. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-SQL one-way receive port, which polls the SQL Server database using the statements specified for the **PollingStatement** binding property, is running.
- The FILE send port, which receives messages from SQL Server, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** on the Employee table and determines that the table has records for polling.
- The adapter executes the polling statement. Because the polling statement consists of a SELECT statement and stored procedures, the adapter will execute all the statements one after the other.
 - The adapter first executes the SELECT statement that returns all the records in the Employee table.
 - The adapter then executes the MOVE_EMP_DATA stored procedure that moves all data from the Employee table to the EmployeeHistory table. This stored procedure does not return any value.
 - The adapter then executes the ADD_EMP_DETAILS stored procedure that adds one record to the Employee table. This stored procedure returns the Employee ID for the inserted record.

So, the message received from SQL Server will contain multiple result sets (for SELECT statement and for ADD_EMP_DETAILS stored procedure), and will resemble the following:

```

<?xml version="1.0" encoding="utf-8" ?>
<Polling xmlns="http://schemas.microsoft.com/Sql/2008/05/Polling/">
  <PolledData>
    <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data">
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" name="Employee_ID" type="xs:int" />
                    <xs:element minOccurs="0" name="Name" type="xs:string" />
                    <xs:element minOccurs="0" name="DOJ" type="xs:dateTime" />
                    <xs:element minOccurs="0" name="Designation" type="xs:string" />
                    <xs:element minOccurs="0" name="Job_Description" type="xs:string" />
                    <xs:element minOccurs="0" name="Photo" type="xs:base64Binary" />
                    <xs:element minOccurs="0" name="Rating" type="xs:string" />
                    <xs:element minOccurs="0" name="Salary" type="xs:decimal" />
                    <xs:element minOccurs="0" name="Last_Modified" type="xs:base64Binary" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
      <diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
        <NewDataSet xmlns="">
          <NewTable>
            <Employee_ID>10001</Employee_ID>
            <Name>John</Name>
            <Designation>Tester</Designation>
            <Salary>100000.00</Salary>
            <Last_Modified>AAAAAAAF34=</Last_Modified>
          </NewTable>
          .....
          .....
          <NewTable>
            <Employee_ID>10005</Employee_ID>
            <Name>Wilson</Name>
            <Designation>Tester3</Designation>
            <Salary>100000.00</Salary>
            <Last_Modified>AAAAAAAF4E=</Last_Modified>
          </NewTable>
        </NewDataSet>
      </diffgr:diffgram>
    </DataSet>
    <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data">
      <xs:schema id="NewDataSet" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
        <xs:element msdata:IsDataSet="true" name="NewDataSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="NewTable">
                <xs:complexType>

```

```

        <xs:sequence>
          <xs:element minOccurs="0" name="Employee_ID" type="xs:int" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
<diffgr:diffgram xmlns:diffgr="urn:schemas-microsoft-com:xml-diffgram-v1">
  <NewDataSet xmlns="">
    <NewTable>
      <Employee_ID>10006</Employee_ID>
    </NewTable>
  </NewDataSet>
</diffgr:diffgram>
</DataSet>
</PolledData>
</Polling>

```

The preceding response contains two data sets. The first data set contains the response for the SELECT statement. The SELECT statement selects all the records in the Employee table. The second data set is for the ADD_EMP_DETAILS stored procedure. This stored procedure adds a record to the Employee table and returns the employee ID for the new record.

Note

The MOVE_EMP_DATA stored procedure does not return a result set. So, there is no corresponding data set in the response message.

- When the adapter executes the **PollDataAvailableStatement** again, it finds one record that was inserted by the ADD_EMP_DETAILS stored procedure. The adapter then executes all three statements that are specified for the **PollingStatement** binding property. This time, the response from SQL Server contains just one record for the SELECT statement and one record for the ADD_EMP_DETAILS stored procedure. All subsequent polls will return similar responses.

Note

The SQL adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling SQL Server by Using the SQL Adapter with BizTalk Server](#)

Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using BizTalk Server

You can configure the SQL adapter to receive strongly-typed polling messages from SQL Server. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set.

You must use strongly-typed polling in a scenario where you want to map the elements in the polling message to any other schema. The schema you want to map to could be for another operation on SQL Server. For example, you could map certain elements in the polling message to the schema for an Insert operation on another table. So, the values in the polling message serve as parameters for the Insert operation. In a simpler scenario, you could map the schema for strongly-typed polling message to a schema file that just stores information.

Important

If you want to have more than one polling operation in a single BizTalk application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. With a unique connection URI, you can create multiple receive ports that poll the same database, or even the same table in a database. For more information, see [Receiving Polling Messages Across Multiple Receive Ports](#).

For more information about how the adapter supports strongly-typed polling, see [Support for Polling](#). For more information about the message schema for strongly-typed polling, see [Message Schemas for the Polling and TypedPolling Operations](#).

How this Topic Demonstrates Strongly-typed Polling

This topic demonstrates how to use strongly-typed polling to map the polling message to another schema. This topic shows how to create a BizTalk project and generate schema for **TypedPolling** operation. Before generating schema for **TypedPolling** operation, you must do the following:

- You must specify an **InboundID** as part of the connection URI.
- You must specify a polling statement for the **PollingStatement** binding property.

As part of the polling statement, perform the following operations:

- Select all the rows from the Employee table.
- Execute a stored procedure (MOVE_EMP_DATA) to move all the records from the Employee table to an EmployeeHistory table.
- Execute a stored procedure (ADD_EMP_DETAILS) to add a new record to the Employee table. This procedure takes the employee name, designation, and salary as parameters.

To perform these operations, you must specify the following for the **PollingStatement** binding property:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000
```

Because you generate schema for the **TypedPolling** operation, the schema is strongly-typed and contains all the elements that will be included in the polling message.

As part of the same BizTalk project, you add another schema file, for example EmployeeDetails.xsd. The schema for EmployeeDetails.xsd resembles the following:

```
<?xml version="1.0" encoding="utf-16" ?>
<xs:schema xmlns:b="http://schemas.microsoft.com/BizTalk/2003" xmlns="http://Typed_Polling.
EmployeeDetails" elementFormDefault="qualified" targetNamespace="http://Typed_Polling.Empl
oyeeDetails" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="EmployeeDetails">
```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="Employee_Info" type="xs:string" />
    <xs:element name="Employee_Profile" type="xs:string" />
    <xs:element name="Employee_Performance" type="xs:string" />
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

You also add a BizTalk Mapper to the project to map the elements from the Employee table (received as polling message) to the elements in the EmployeeDetails.xsd schema. As part of the map, you combine one or more elements from the polling message and map it to a single element in the EmployeeDetails schema. You can do so by using the **String Concatenate** functoid.

Finally, as part of the BizTalk project, a file conforming to the EmployeeDetails.xsd schema is dropped to a FILE send port.

Configuring Typed Polling with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure the adapter to receive data-change messages. Other than the **PollingStatement** binding property, all the other binding properties listed in this section are required while configuring the receive port in the BizTalk Server Administration console. You must specify the **PollingStatement** binding property before generating schema for the **TypedPolling** operation.

 **Note**

For typed polling, you must specify the **PollingStatement** binding property while generating the schema. You can choose to specify the other binding properties as well while generating the schema, even though they are not mandatory. If you do specify the binding properties, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-SQL receive port with the binding properties already set. For more information about creating a port using the binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling , TypedPolling , or Notification inbound operation. Default is Polling . To receive strongly-typed polling messages, set this to TypedPolling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. The SQL statement must return a result set consisting of rows and columns. Only if a row is available, the SQL statement specified for the PollingStatement binding property will be executed.
PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter waits for the remaining time in the interval.
PollingStatement	Specifies the SQL statement to poll the SQL Server database table. You can specify a simple SELECT statement or a stored procedure for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property. You can specify any number of SQL statements separated by a semi-colon.  Important For TypedPolling , you must specify this binding property before generating metadata.

PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false .
---------------------------	--

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to poll SQL Server, read further.

How to Receive Strongly-typed Data-change Messages from the SQL Server Database

Performing an operation on the SQL Server database using SQL adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive strongly-typed data-change messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **TypedPolling** operation. You must specify the **InboundID** connection property and the **PollingStatement** binding property while generating schema. For example, a connection URI with the inbound ID specified resembles the following:

```
mssql://mysqlserver//mysqldatabase?InboundID=mydatabaseId
```

2. Create a message in the BizTalk project for receiving messages from the SQL Server database.
3. Create an orchestration to receive messages from the SQL Server database and to save them to a folder.
4. Add a schema, for example, EmployeeDetails.xsd, in the BizTalk project.
5. Add a BizTalk Mapper to map the schema for the polling message to EmployeeDetails.xsd schema.
6. Build and deploy the BizTalk project.
7. Configure the BizTalk application by creating physical send and receive ports.

◆ Important

For inbound polling scenarios you must always configure a one-way WCF-Custom or WCF-SQL receive port. Two-way WCF-Custom or WCF-SQL receive ports are not supported for inbound operations.

8. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based on This Topic

A sample, TypedPolling, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

You must generate the schema for the **TypedPolling** operation. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema.

1. Specify the **InboundID** connection property while specifying the connection URI. For this topic, you can specify the **InboundID** as **Employee**. For more information about the connection URI, see [SQL Server Connection URI](#).
2. Specify a value for the **PollingStatement** binding property. For more information about this binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

For instructions on how to specify binding properties, see [Specifying Binding Properties](#).

3. Select the contract type as **Service (Inbound operation)**.

4. Generate schema for the **TypedPolling** operation.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from the SQL Server database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type PollingMessage .
Message Type	From the drop-down list, expand Schemas , and select <i>Typed_Polling.TypedPolling_Employee.TypedPolling</i> , where <i>Typed_Polling</i> is the name of your BizTalk project. <i>TypedPolling_Employee</i> is the schema generated for the Typed Polling operation.

Setting up the Orchestration

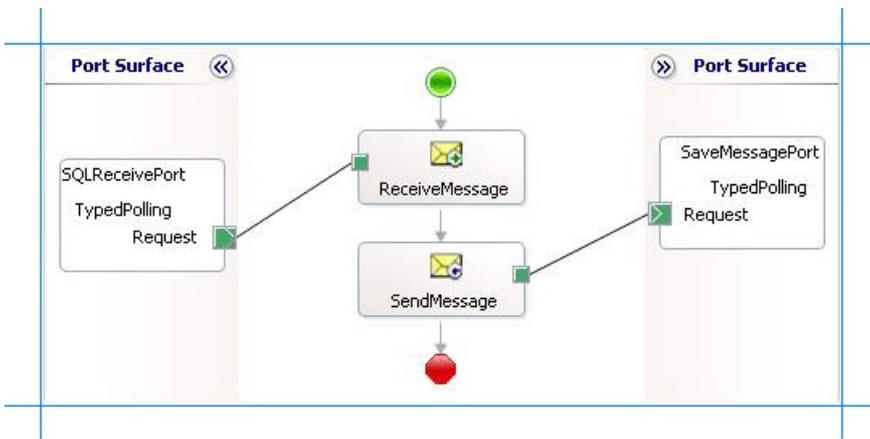
You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from the SQL Server database. In this orchestration, the adapter receives the polling message for the specified polling statement. The BizTalk Mapper then maps the polling message schema to the EmployeeDetails.xsd schema. The mapped message is then saved to a FILE location. A typical orchestration for receiving strongly-typed polling message from a SQL Server database would contain:

- Receive and Send shapes to receive messages from SQL Server and send to a FILE port, respectively.
- A one-way receive port to receive messages from SQL Server.

◆ Important
For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from a SQL Server database to a folder.
- A BizTalk Mapper to map the schema of the polling message to any other schema.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveMessage	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveMessage</i> Set Activate to <i>True</i>
SaveMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SaveMessage</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
SQLReceivePort	<ul style="list-style-type: none"> Set Identifier to <i>SQLReceivePort</i> Set Type to <i>SQLReceivePortType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Receive</i>
SaveMessagePort	<ul style="list-style-type: none"> Set Identifier to <i>SaveMessagePort</i> Set Type to <i>SaveMessagePortType</i> Set Communication Pattern to <i>One-Way</i> Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> Set Message to <i>PollingMessage</i> Set Operation to <i>SQLReceivePort.TypedPolling.Request</i>

SaveMessage	<ul style="list-style-type: none"> • Set Message to <i>PollingMessage</i> • Set Operation to <i>SaveMessagePort.TypedPolling.Request</i>
-------------	--

After you have specified these properties, the message shapes and ports are connected.

Add a BizTalk Mapper

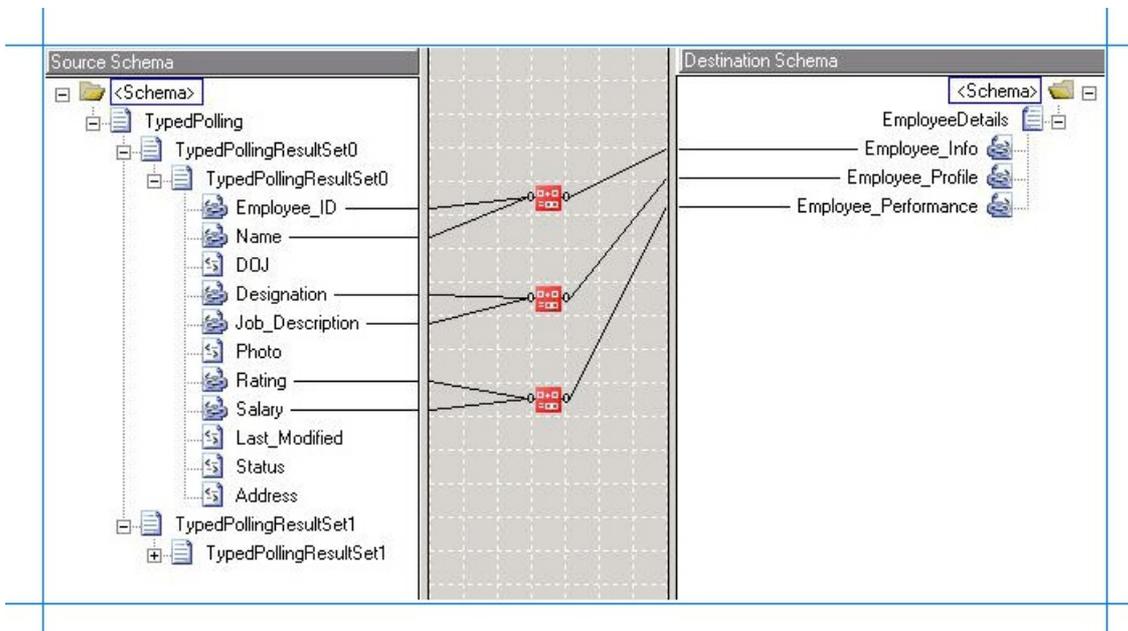
You must add a BizTalk Mapper to the orchestration to map the polling message schema to the EmployeeDetails.xsd schema. In the BizTalk Server Administration console, you will use this Mapper to map the schema for the polling message to the EmployeeDetails.xsd schema.

To add a BizTalk Mapper

1. Add a BizTalk Mapper to the BizTalk project. Right-click the BizTalk project, point to **Add**, and click **New Item**.
In the **Add New Item** dialog box, from the left pane, select **Map Files**. From the right pane, select **Map**. Specify a name for the map, such as **MapSchema.btm**. Click **Add**.
2. From the Source Schema pane, click **Open Source Schema**.
3. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for the polling message. For this topic, select *Typed_Polling.TypedPolling_Employee*. Click **OK**.
4. In the **Root Node for Source Schema** dialog box, select *TypedPolling* and click **OK**.
5. From the Destination Schema pane, click **Open Destination Schema**.
6. In the **BizTalk Type Picker** dialog box, expand the project name, expand **Schemas**, and select the schema for EmployeeDetails. For this topic, select *Typed_Polling.EmployeeDetails*. Click **OK**.
7. In the source schema of polling message, expand the *TypedPollingResultSet0* node and the subsequent nodes to see the elements that are returned in the polling message. In the destination schema, expand the *EmployeeDetails* node to see the different elements in the schema. For this topic, you must map the schemas in such a way that:
 - **Employee_ID** and **Name** in the source schema must map to **Employee_Info** in the destination schema.
 - **Designation** and **Job_Description** in the source schema must map to **Employee_Profile** in the destination schema.
 - **Rating** and **Salary** in the source schema must map to **Employee_Performance** in the destination schema.

To combine more than one node in source schema and map them to a single node in the destination schema, you must use the **String Concatenate** functoid. For more information about the functoid, see <http://go.microsoft.com/fwlink/?LinkId=123474>.

8. To use the String Concatenate functoid:
 - a. From the **Toolbox**, drag the **String Concatenate** functoid and drop it on the Mapper grid.
 - b. Connect the **Employee_ID** and **Name** elements in the source schema to the functoid.
 - c. Connect the functoid to the **Employee_Info** element in the destination schema.
 - d. Repeat these steps for all the elements that you want to map. A finished map will resemble the following:



e. Save the map.

The orchestration is complete after you create the Mapper. You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-SQL one-way receive port. This port polls the SQL Server database with the polling statement you specify for the port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Make sure you specify the following binding properties for the receive port.

◆ Important

Make sure you specify the **InboundID** as part of the connection URI. The inbound ID must be the same you specified while generating the schema.

◆ Important

You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-SQL receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Make sure you set this to TypedPolling .

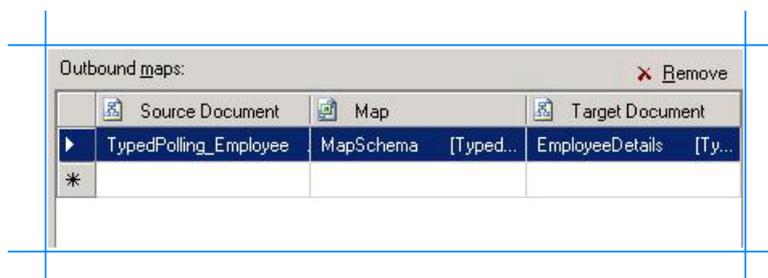
PolledDataAvailableStatement	<p>Make sure you specify the same SQL statement you specified while generating the schema, which is:</p> <pre>SELECT COUNT(*) FROM Employee</pre>
PollingStatement	<p>Make sure you provide the same polling statement you specified while generating the schema, which is:</p> <pre>SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000</pre>

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note

We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the SQL adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-SQL receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a FILE send port where the adapter will drop the message. This send port will also use the map you created in the orchestration to map the polling message to a message conforming to the EmployeeDetails.xsd schema. Perform the following steps to configure the FILE send port to use the map.
 1. Create a FILE send port.
 2. From the left pane of the send port properties dialog box, click **Outbound Maps**. From the right pane, click the field under the **Map** column, and from the drop-down select **MapSchema**. Click **OK**.



Starting the Application

You must start the BizTalk application for receiving messages from the SQL Server database. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-SQL one-way receive port, which polls the SQL Server database using the statements specified for the **PollingStatement** binding property, is running.
- The FILE send port, which will map the polling message to the EmployeeDetails schema, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** on the Employee table and determines that the table has

records for polling.

- The adapter executes the polling statement. Because the polling statement consists of a SELECT statement and stored procedures, the adapter will execute all the statements one after the other.
 - The adapter first executes the SELECT statement that returns all the records in the Employee table.
 - The adapter then executes the MOVE_EMP_DATA stored procedure that moves all data from the Employee table to the EmployeeHistory table. This stored procedure does not return any value.
 - The adapter then executes the ADD_EMP_DETAILS stored procedure that adds one record to the Employee table. This stored procedure returns the Employee ID for the inserted record.

After the polling statement is executed and the message is received, the polling message gets sent to the FILE send port. Here, the outbound map (**MapSchema**) configured on the send port maps the polling message to the EmployeeDetails schema and drops the message to a file location. The message resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<EmployeeDetails xmlns="http://Typed_Polling.EmployeeDetails">
  <Employee_Info>10751John</Employee_Info>
  <Employee_Profile>TesterManagesTesting</Employee_Profile>
  <Employee_Performance>100000</EmployeePerformance>
</EmployeeDetails>
```

In the preceding response, you can notice that the Employee_Info element contains a combination of employee ID (10751) and employee name (John). The other elements also contain combinations as mapped in the Mapper you created as part of the orchestration.

Note

The SQL adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling SQL Server by Using the SQL Adapter with BizTalk Server](#)

Receiving Polling Messages Using SELECT Statements with FOR XML Clause

You can configure the SQL adapter to receive periodic data-change messages for SQL Server tables or views by using SELECT statements or stored procedures that include a FOR XML clause. You can specify these statements as polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set.

For more information on how the adapter supports polling, see [Support for Polling](#). For information about the structure of the SOAP message for polling operations, see [Message Schemas for the Polling and TypedPolling Operations](#). For more information about the FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

Note

This topic demonstrates how to use the **XmlPolling** inbound operation to receive polling messages. The **XmlPolling** operation is used to poll a SQL Server database using SELECT statements or stored procedures that include a FOR XML clause. The message for the **XmlPolling** operation includes the xml message received by executing the SELECT statement or the stored procedure in SQL Server Management Studio.

You can also use the adapter to receive different types of polling messages.

- If you want to get a weakly-typed polling message, you must use the Polling Operation. For more information, see [Receiving Polling-based Data-changed Messages from SQL Server by Using BizTalk Server](#).
- If you want to get strongly-typed polling message, you must use the **TypedPolling** operation. You must also use the **TypedPolling** operation to have multiple polling operations in a single BizTalk application. For instructions on how to perform **TypedPolling** operation, see [Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using BizTalk Server](#).

Important

If you want to have more than one polling operation in a single BizTalk application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. With a unique connection URI, you can create multiple receive ports that poll the same database, or even the same table in a database. For more information, see [Receiving Polling Messages Across Multiple Receive Ports](#).

How This Topic Demonstrates Polling

In this topic, to demonstrate how the SQL adapter supports receiving data change messages, we use a SELECT statement with the FOR XML clause to poll the SQL Server database. When you invoke such a statement in SQL Server Management Studio, the output is in the form of an xml message. To use such statements to poll a SQL Server database, you must have the schema of the response xml message. The SQL adapter requires this schema to receive a polling message after executing a SELECT statement with the FOR XML clause. So, to use a SELECT statement with FOR XML clause to poll the SQL Server database, you must perform the following set of tasks.

1. Generate the schema for the XML response message for the SELECT statement with FOR XML clause.
2. Create a BizTalk project and add the generated schema to the project.
3. Create a message in the BizTalk project for receiving XML response messages from the SQL Server database.
4. Create an orchestration to receive messages from the SQL Server database and to save them to a folder.
5. Build and deploy the BizTalk project.
6. Configure the BizTalk application by creating physical send and receive ports.

Important

For inbound polling scenarios you must always configure a one-way WCF-Custom or WCF-SQL receive port. Two-way WCF-Custom or WCF-SQL receive ports are not supported for inbound operations.

7. Start the BizTalk application.

Generating Schema for the Response Message for SELECT Statement

You can generate the schema for the response message for the SELECT statement by including the `xmlschema` clause with the `for xml` clause. In this topic, we use a SELECT statement to retrieve employee details for a given employee ID. To retrieve the schema by executing a SELECT statement, the SELECT statement must be written in the following manner:

```
SELECT Employee_ID ,Name ,Designation FROM Employee for xml auto, xmlschema
```

Execute this SELECT statement to get the schema for the response message. Save the schema. You must now create a BizTalk project in Visual Studio and add this schema to the project. For this example, you can name this schema as **PollingResponse.xsd**.

◆ Important

Make sure you remove the `xmlschema` clause after you have executed the SELECT statement to generate the schema. If you fail to do this, when you finally execute the SELECT statement through BizTalk as part of the XmlPolling operation, you will again generate the schema in the response message. So, to get the response message as xml you must remove the `xmlschema` clause.

To add the schema to a BizTalk project

1. Create a BizTalk project in Visual Studio.
2. Add the response schema you generated for the stored procedure to the BizTalk project. Right-click the BizTalk project in the Solution Explorer, point to **Add**, and then click **Existing Item**. In the Add Existing Item dialog box, navigate to the location where you saved the schema and click **Add**.
3. Open the schema in Visual Studio and make the following changes.
 - a. Add a node to the schema and move the existing root node under this newly added node. Give a name to the root node. For this topic, rename the root node to **Root**.
 - b. The response schema generated for the SELECT statement references a `sqltypes.xsd`. You can get the `sqltypes.xsd` schema from <http://go.microsoft.com/fwlink/?LinkId=131087>. Add the `sqltypes.xsd` schema to the BizTalk project.
 - c. In the schema generated for the SELECT statement, change the value of `import schemaLocation` to the following.

```
import schemaLocation="sqltypes.xsd"
```

You do this because you have already added the `sqltypes.xsd` schema to your BizTalk project.

- d. Provide a target namespace for the schema. Click the **<Schema>** node, and in the properties pane, specify a namespace in the **Target Namespace** property. For this topic, give the namespace as `http://ForXmlPolling/namespace`.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive messages from the SQL Server database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type PollingMessage .
Message Type	From the drop-down list, expand Schemas , and select <i>ForXMLPolling.PollingResponse</i> , where <i>ForXMLPolling</i> is the name of your BizTalk project. <i>PollingResponse</i> is the name of the response schema generated by executing the SELECT statement as described under Generating Schema for the Response Message for SELECT Statement .

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving polling-based data-change messages from the SQL Server database. In this orchestration, the adapter receives the response of the select statement specified for the **PollingStatement** binding property. The response for the SELECT statement is saved to a FILE location. A typical orchestration for polling a SQL Server database would contain:

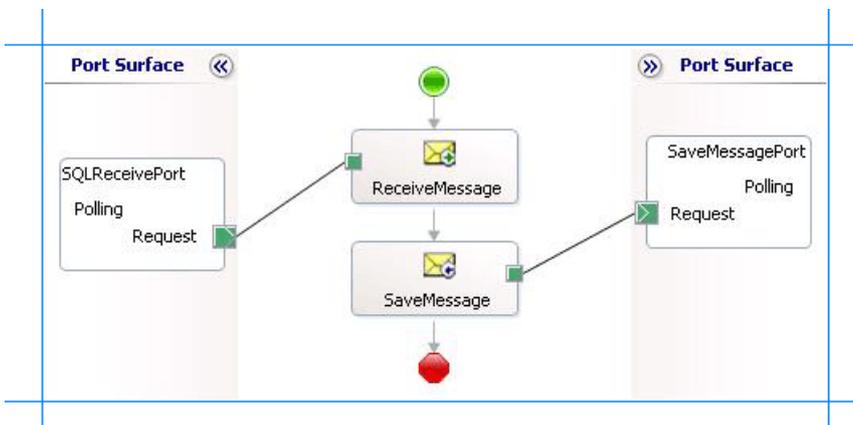
- Receive and Send shapes to receive messages from SQL Server and send to a FILE port, respectively.
- A one-way receive port to receive messages from SQL Server.

◆ Important

For inbound polling scenarios you must always configure a one-way receive port. Two-way receive ports are not supported for inbound operations.

- A one-way send port to send polling responses from a SQL Server database to a folder.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
-------	------------	------------

ReceiveMessage	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveMessage</i> • Set Activate to <i>True</i>
SaveMessage	Send	<ul style="list-style-type: none"> • Set Name to <i>SaveMessage</i>

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
SQLReceivePort	<ul style="list-style-type: none"> • Set Identifier to <i>SQLReceivePort</i> • Set Type to <i>SQLReceivePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SQLReceivePort.XmlPolling.Request</i>
SaveMessage	<ul style="list-style-type: none"> • Set Message to <i>Receive</i> • Set Operation to <i>SaveMessagePort.XmlPolling.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the messages from the SQL Server database. These messages will be in response to the polling statement that you specify for the receive port.
 - Define a physical WCF-Custom or WCF-SQL one-way receive port. This port polls the SQL Server database with the polling statement you specify for the port. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Make sure you specify the following binding properties for the receive port.

◆ Important
 You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-SQL receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Make sure you set this to XmlPolling .
PolledDataAvailableStatement	Make sure you specify a SQL statement. For this topic, specify: <pre>SELECT COUNT(*) FROM Employee</pre>
PollingStatement	Make sure you provide the same statement, without the <code>xmlschema</code> clause, that you specified while generating the schema as described in Generating Schema for the Response Message for SELECT Statement . <pre>SELECT Employee_ID ,Name ,Designation FROM Employee for xml auto</pre> <p>Note Note that the SELECT statement does not contain the <code>xmlschema</code> clause.</p>
XmlStoredProcedureRootNodeName	Specify the name of the root node that you added to the response schema you generated for the SELECT statement, as described under Generating Schema for the Response Message for SELECT Statement . For this topic, set this to Root .
XmlStoredProcedureRootNamespace	Specify the target namespace for the response schema you generated for the SELECT statement, as described under Generating Schema for the Response Message for Stored Procedure . For this topic, set this to <code>http://ForXmlPolling/namespace</code> .

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

◆ Note
 We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the SQL adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-SQL receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

Starting the Application

You must start the BizTalk application for receiving messages from the SQL Server database. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-SQL one-way receive port, which polls the SQL Server database using the statements specified for the **PollingStatement** binding property, is running.
- The FILE send port, which receives messages from SQL Server, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you run the application, the following set of actions take place, in the same sequence:

- The adapter executes the **PolledDataAvailableStatement** on the Employee table and determines that the table has records for polling.
- The adapter executes the polling statement and receives a polling message from the SQL Server database. Because the polling statement consists of a SELECT statement with a FOR XML clause, the polling message received by the adapter resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Root xmlns="http://ForXmlPolling/namespace">
  <Employee Employee_ID="10765" Name="John" Designation="Tester" xmlns="" />
  <Employee Employee_ID="10766" Name="Sam" Designation="Manager" xmlns="" />
  .....
  .....
</Root>
```

Notice that the polling message is received in the same schema as generated by executing the SELECT statement with the **xmlschema** clause. Also note that the root node and the namespace is the same you specified as values for **XmlStoredProcedureRootNodeName** and **XmlStoredProcedureRootNodeNamespace** binding properties respectively.

Note

The SQL adapter will continue to poll until you explicitly disable the receive port from the BizTalk Server Administration console.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Polling SQL Server by Using the SQL Adapter with BizTalk Server](#)

Receiving Polling Messages Across Multiple Receive Ports

Consider a scenario where you want to create a BizTalk application that includes two polling operations. Each polling operation polls separate tables, Employee and Customer, from the same database. When you deploy such an application in BizTalk Server Administration console, you will need to create two receive ports. The connection URI for each receive port will be:

```
mssql://<server_name>/<database_instance_name>/<database_name>
```

Because both receive ports are receiving polling messages from the same database on the same server, the connection URI for both will be the same. However, a BizTalk application cannot have two receive ports with the same connection URI.

To enable adapter clients to have two receive ports that poll the same database (or even the same table in a database) in a BizTalk application, the SQL adapter provides a connection property, **InboundID**. You can specify any value for this connection property. By adding the inbound ID, a connection URI becomes unique. For example:

The connection URI for the port receiving polling messages for the Employee table can be:

```
mssql://<server_name>/<database_instance_name>/<database_name>?InboundID=Employee
```

Similarly, the connection URI for the port receiving polling messages for the Customer table can be:

```
mssql://<server_name>/<database_instance_name>/<database_name>?InboundID=Customer
```

Because the connection URIs become unique by adding the **InboundID** property, you can now have multiple receive ports polling the same database or table in a single BizTalk application.

◆ Important

You can choose to specify the **InboundID** connection property for both the **Polling** and **TypedPolling** operations.

See Also

Other Resources

[Polling SQL Server by Using the SQL Adapter with BizTalk Server](#)

Receiving Query Notifications by Using BizTalk Server

You can configure the SQL adapter to receive notification messages for SQL Server tables or views. You can specify a SQL statement that the adapter uses to register for notifications with SQL Server. The notification statement can be a SELECT statement or a stored procedure that returns a result set. For more information about query notifications, see "Using Query Notifications" at <http://go.microsoft.com/fwlink/?LinkId=122159>. For information about queries that can be used for query notifications, see "Creating a Query for Notification" at <http://go.microsoft.com/fwlink/?LinkId=122160>.

Receiving query notifications from SQL Server is similar to polling SQL Server, with a few key differences. For the list of differences, see [Receiving Query Notifications](#).

Following are some scenarios in which you can configure the SQL adapter with BizTalk Server to receive notifications from SQL Server:

- Adapter clients get only "incremental" notification, for example, only for those changes that were made to a database table since the last notification.
- If many rows are inserted into a database table, the adapter clients can configure multiple receive locations to load-balance receiving notifications.
- If the receive location on which the adapter clients are receiving notifications goes down, the adapter clients can configure the adapter to receive a notification as soon as the receive location is up again. The clients must also implement the logic in their application to process the records that may have been inserted, updated, or deleted while the receive location was down.

Once the adapter clients receive a notification message, they can perform specific tasks based on the kind of notification received. For example, a BizTalk orchestration can be designed in such a way that it performs one set of tasks if an insert notification is received and another set of tasks if an update notification is received.

The topics in this section provide information about how to configure the adapter for each of these scenarios. To start getting notifications from SQL Server using the SQL adapter, you must specify certain binding properties. For more information about how the adapter supports receiving messages, see [Receiving Query Notifications](#). For more information about the binding properties related to notifications, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For more information about the structure of notification messages, see [Message Schemas for Query Notification](#).

You must also perform the following tasks on SQL Server to enable query notifications.

- You must enable Service Broker for the SQL Server database.
- You must ensure that the adapter client has the necessary permissions to execute commands to request notification.

For more information about these tasks, see "Enabling Query Notifications" at <http://go.microsoft.com/fwlink/?LinkID=122323>.

In This Section

- [Considerations for Receiving Query Notifications Using the Adapter](#)
- [Processing Notification Messages to Perform Specific Tasks](#)
- [Receiving Query Notifications Incrementally Using BizTalk Server](#)
- [Receiving Query Notifications On Multiple Receive Locations](#)
- [Receiving Query Notifications After a Receive Location Breakdown](#)

See Also

Other Resources

[Developing BizTalk Applications](#)

Considerations for Receiving Query Notifications Using the Adapter

This topic provides some considerations and best practices to keep in mind while using the SQL adapter to receive query notifications from a SQL Server database.

Considerations While Using the Adapter to Receive Notifications

You must consider the following while using the SQL adapter to receive query notifications:

- The SQL adapter receives the query notification from SQL Server, and then simply passes on the notification to the adapter clients. The adapter does not distinguish between the notifications for different operations (i.e., the adapter does not have any information whether a particular notification is for an Insert operation or an Update operation).
- The notification message for an operation is not affected by the number of records affected by that operation. For example, regardless of the number of records inserted, updated, or deleted in a SQL Server database table, the adapter client receives only one notification message.
- We recommend that the adapter client application contain the logic to interpret the type of notification received from SQL Server. The notification type can be determined by extracting the information from, the **<Info>** element of the received notification message. Here's an example of a notification message received for an Insert operation:

```
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Insert</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

Notice the value within the **<Info>** element. This value provides information on the operation for which the notification message was received. Your application should have the functionality to extract the value within the **<Info>** element and then based on the value, perform subsequent tasks. The topic [Processing Notification Messages to Perform Specific Tasks](#) has instructions on how to extract the value within the **<Info>** element. A detailed tutorial that performs similar tasks is also available at [Tutorial 2: Employee - Purchase Order Process](#).

- Ideally, after the client application receives a notification for a specific record, that record should be updated so that additional notifications are not received. For example, consider an **Employee** table that has a **Status** column. For all new records inserted into the **Employee** table, the value in the **Status** column is always "0" so the table will look like the following:

Employee Name	Status
John	0

To receive notifications for the newly inserted record, the adapter client will set the **NotificationStatement** binding property as:

```
SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0
```

Note

You must specifically specify the column names in the statement as shown in this SELECT statement. Also, you must always specify the table name along with the schema name. For example, `dbo.Employee`.

After receiving the notification, the client application must reset the value of the **Status** column to "1" so that the notification statement does not operate on the record again. To achieve this, the client application must perform an Update operation on the **Employee** table. After the Update operation, the same record in the **Employee** table will look

like the following:

Employee Name	Status
John	1

Interestingly, the Update operation will again send a notification to the adapter client and the whole process will be repeated again, therefore, the client application must have the required logic to discard such unwanted notifications.

- If the **NotifyOnListenerStart** binding property is true, the adapter client will receive a notification message every time the receive location starts. For more information on how to use the binding property and interpret the notification message, see [Receiving Query Notifications After a Receive Location Breakdown](#).

Typical Orchestration for Receiving Notifications

This section outlines the typical orchestration flow for receiving notifications using the SQL adapter.

1. The first thing that the orchestration must do is to determine the type of notification received, including:

- Whether the notification was received after a receive location restarts.
- Whether the notification was received for an operation on a database table, such as Insert, Update, or Delete.

The orchestration must include an **Expression** shape, and within that, an xpath query to decide what kind of message is received.

2. After the notification type is determined, the orchestration must include a decision block to perform specific actions based on the type of notification received. To achieve this, the orchestration must include a **Decide** shape, which includes a **Rule** block and an **Else** block:

- Within the **Rule** block, you must specify the condition and then include orchestration shapes to perform certain operations if the condition is met.
- Within the **Else** block, you must include orchestration shapes to perform certain operations if the condition is *not* met.

Details of the preceding requirements are described in [Processing Notification Messages to Perform Specific Tasks](#). A detailed tutorial is also available in [Tutorial 2: Employee - Purchase Order Process](#).

Processing Notification Messages to Perform Specific Tasks

You can use the SQL adapter to receive notifications for changes to SQL Server database tables. However, the adapter only sends you a notification that some records were inserted, updated, or deleted in a certain database table. Any post-processing on those records must be handled by the client applications themselves. This topic presents a scenario-based description on how to process the records in the table based on the kind of notification received from the SQL Server database.

Scenarios for Performing Subsequent Actions After Receiving Notification

Following are a couple of scenarios in which the adapter clients must perform certain post-notification tasks.

- **Scenario 1.** Consider a scenario where the adapter client must perform certain tasks based on the kind of notification you receive from SQL Server. For example, the client application must update the records in table "A" if records are inserted in table "B". Similarly, the client application must delete records from table "A" if records are deleted from table "B".

In this scenario, from the notification message received, the adapter clients must extract the type of notification to decide whether the notification was for an insert operation or a delete operation. Once the notification type is ascertained, the adapter clients must perform subsequent actions to insert or update the relevant tables.

- **Scenario 2.** Consider a scenario where the receive location that receives notification messages for changes to a table goes down. While the receive location is down, some records are added to the table. However, for these records the adapter client does not receive any notification. When the receive location is back up, the adapter notifies the client by sending a specific message and then the client application must look for all the records that were inserted in the database table while the receive location was down.

In this scenario, from the notification message received, the adapter clients must extract the information regarding whether the notification is for a change to a database table or for the receive location starting. If the notification is for the receive location starting, the adapter clients must implement the logic to process the records that might have been inserted, updated, or deleted while the receive location was down.

Note

These are just some example scenarios that are listed for a better understanding of how to use the notification feature in the SQL adapter. However, the basic set of tasks required to extract the type of notification received will be similar for all scenarios. This topic provides instructions on how to extract the type of notification from a notification message.

How This Topic Demonstrates Receiving Notification Messages and Extracting Notification Type

In this topic, to demonstrate how to process notification messages to perform subsequent tasks, we consider a basic scenario where an adapter client uses BizTalk application to receive notification messages for changes to the Employee table. After the notification is received, the client filters the type of notification received and performs subsequent action. To demonstrate a very basic scenario, let us consider that the adapter client copies the notification messages to different folders based on the kind of notification received. So:

- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.
- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotification folder.

To achieve this as part of a BizTalk application, the orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the information about the kind of notification message received.

- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.

Configuring Notifications with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure receiving notifications from SQL Server. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

Note
You may choose to specify these binding properties when generating the schema for the Notification operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-SQL receive port with the binding properties already set. For more information about creating a port using the binding file, see Configuring a Physical Port Binding Using a Port Binding File .

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for query notifications. The adapter gets a notification message from SQL Server only when the result set for the specified SQL statement changes.
NotifyOnListenerStart	Specifies whether adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to receive notifications from SQL Server, read further.

How to Receive Notification Messages from the SQL Server Database

Performing an operation on the SQL Server database using SQL adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** inbound operation. Optionally, you can specify values for the **InboundOperationType** and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from the SQL Server database.
3. Create an orchestration as described in the preceding section.
4. Build and deploy the BizTalk project.
5. Configure the BizTalk application by creating physical send and receive ports.

Note
For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-SQL receive port. Two-way WCF-Custom or WCF-SQL receive ports are not supported for inbound operations.

6. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Generating Schema

You must generate the schema for the **Notification** inbound operation. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design-time.

1. Specify a value for **InboundOperationType** and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create one message to receive notifications from the SQL Server database.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>Process_Notification.Notification</i> , where <i>Process_Notification</i> is the name of your BizTalk project. <i>Notification</i> is the schema generated for the Notification operation.

Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from the SQL Server database and then performing tasks based on the type of notification received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The xpath query specified within the Expression shape extracts the type of notification into a variable, say **NotificationType**. The Decide shape uses the value in this variable to decide on the kind of notification received and takes the appropriate "path" to perform subsequent operations. As mentioned in the preceding section, the orchestration will perform the following operations based on the kind of notification message received.

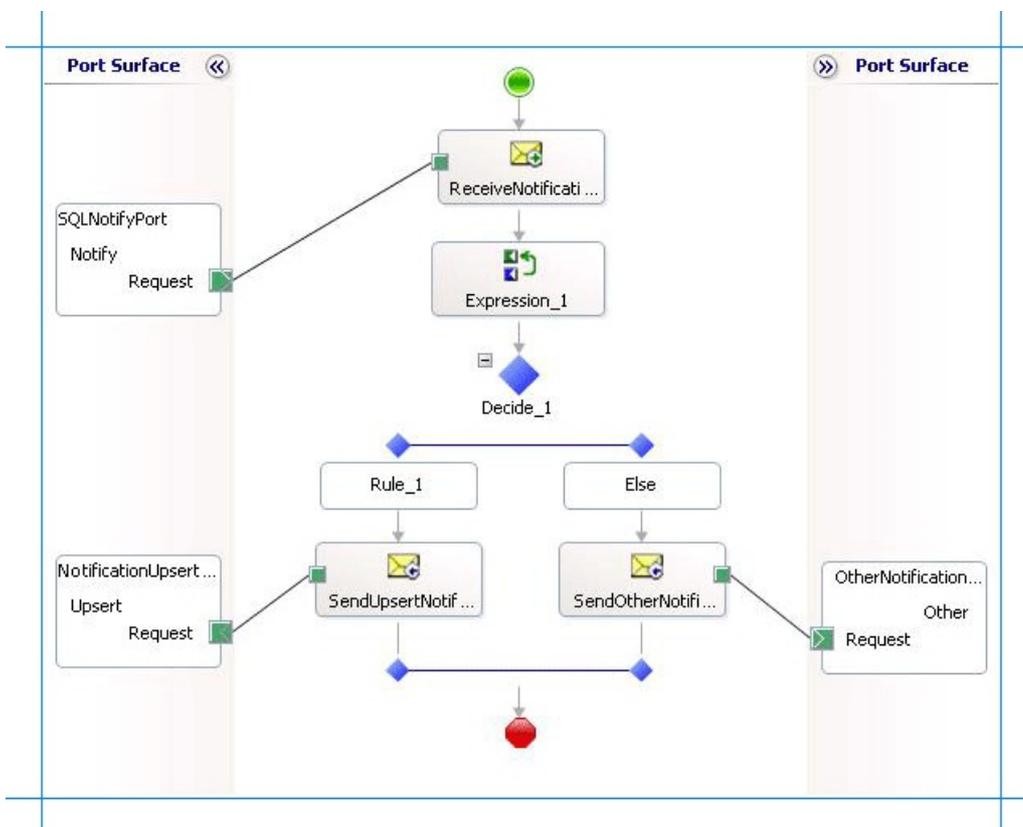
- If the notification message is for an Insert or Update operation, the adapter client copies the message to C:\TestLocation\UpsertNotification folder.

- If the notification message is for any other operation, for example Delete, the adapter client copies the message to C:\TestLocation\OtherNotificaiton folder.

So, your orchestration must contain the following:

- A one-way receive port to receive notification messages.
- An Expression shape that contains an xpath query to extract the kind of notification received.
- A Decide shape to include a decision block in the orchestration. In this decision block, the application decides on what subsequent operations to perform based on the notification message received.
- Two one-way send ports that finally receive the notification messages.
- Receive shape.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> • Set Name to <i>ReceiveNotification</i> • Set Activate to <i>True</i>

Adding an Expression Shape

The purpose of including an Expression shape in the orchestration is to have an xpath query to extract the kind of notification message received. Before creating an xpath query, let us look at the format of a notification message. A typical notification message resembles the following:

```
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
```

```
<Info>Insert</Info>
<Source>Data</Source>
<Type>Change</Type>
</Notification>
```

As you see, the information about the type of the notification is available within the `<info>` tag, within the parent `<Notification>` tag. So, as part of this expression shape you must:

- Create a variable that contains the value within the `<Info>` tag and set its type to `System.String`. For more information about creating variables, see <http://go.microsoft.com/fwlink/?LinkId=102357>.

For this topic, name the variable as **NotificationType**.

- Create an xpath query to extract the value from the `<Info>` tag. The xpath query will resemble the following:

```
NotificationType = xpath(NotifyReceive, "string(/*[local-name()='Notification']/*[local-name()='Info']/text())");
```

In this xpath query, **NotifyReceive** is the message you created for receiving notification messages. The excerpt within the `string` function indicates that the query must extract the value within the `<Info>` tag, which in turn is within the `<Notification>` tag. Finally, the value extracted by the query is assigned to the **NotificationType** variable.

Adding a Decide Shape

The purpose of adding a Decide shape is to include a decision block in the orchestration to decide what subsequent operations to perform based on the kind of notification message received. The decision is made on the basis of the value of the **NotificationType** variable. In this topic, the orchestration makes a decision based on the kind of notification message received. So, the condition in the Rule shape is specified as follows:

```
NotificationType.Equals("Insert") | NotificationType.Equals("Update")
```

This condition suggests that if the value for **NotificationType** variable is Insert or Update, the orchestration will perform one set of tasks. If the value of **NotificationType** variable is anything else, the orchestration will perform other set of tasks.

As mentioned in the preceding sections, to demonstrate a simple approach, the orchestration will copy messages to different folders based on the notification message type. So, within the Rule and Else blocks, you must add Send shapes to send the messages to different ports. For this topic, name the Send shape in the Rule block as **SendUperstNotification** and the Send shape in the Else block as **SendOtherNotification**.

Adding Ports

You must now add the following logical ports to the orchestration:

- One-way receive port to receive notification messages from SQL Server.
- One-way send port to send notification messages for Insert and Update operations to a specific folder.
- One-way send port to send notification messages for any other operations to a specific folder.

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
------	------------

SQLNotifyPort	<ul style="list-style-type: none"> • Set Identifier to <i>SQLNotifyPort</i> • Set Type to <i>SQLNotifyPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>
NotificationUpsertPort	<ul style="list-style-type: none"> • Set Identifier to <i>NotificationUpsertPort</i> • Set Type to <i>NotificationUpsertPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>
OtherNotificationPort	<ul style="list-style-type: none"> • Set Identifier to <i>OtherNotificationPort</i> • Set Type to <i>OtherNotificationPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>SQLNotifyPort.Notify.Request</i>
SendUpsertNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>NotificationUpsertPort.Upsert.Request</i>
SendOtherNotification	<ul style="list-style-type: none"> • Set Message to <i>Select</i> • Set Operation to <i>OtherNotificationPort.Other.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkId=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-SQL one-way receive port. This port listens for notifications coming from the SQL Server database. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Make sure you specify the following binding properties for the receive port.

◆Important
 You do not need to perform this step if you specified the binding properties at design-time. In such a case, you can create a WCF-custom or WCF-SQL receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Value
InboundOperationType	Set this to Notification .
NotificationStatement	Set this to: <pre>SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0</pre> <p>Note You must specifically specify the column names in the statement as shown in this SELECT statement. Also, you must always specify the table name along with the schema name. For example, <code>dbo.Employee</code>.</p>
NotifyOnListenerStart	Set this to True .

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Note
 We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the SQL adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-SQL receive port. For instruction on how to add the service behavior, see [Configure Transaction Isolation Level and Transaction Timeout](#).

- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the SQL Server database for Insert and Update operations. Configure this port to drop notification messages to the folder `C:\TestLocation\UpsertNotification`.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the notification messages from the SQL Server database for all other operations. Configure this port to drop notification messages to the folder `C:\TestLocation\OtherNotification`.

Starting the Application

You must start the BizTalk application for receiving notification messages from the SQL Server database and for performing the subsequent Select and Update operations. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-SQL one-way receive port, which receives the notification messages from the SQL Server database is running.
- The two FILE send ports, which receive messages from SQL Server, are running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

After you start the BizTalk orchestration, the following set of actions take place:

- Because the **NotifyOnListenerStart** binding property is set to **True**, you receive the following message:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>ListenerStarted</Info>
  <Source>SqlBinding</Source>
  <Type>Startup</Type>
</Notification>
```

Note that the value in the `<Info>` tag is "ListnerStarted". Hence, this message is received in C:\TestLocation\OtherNotification folder.

- Insert a record in the Employee table. You will receive a notification message resembling the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Insert</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

Note that the value in the `<Info>` tag is "Insert". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Update a record in the Employee table. You will receive a notification message resembling the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Update</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

Note that the value in the `<Info>` tag is "Update". Hence, this message is received in C:\TestLocation\UpsertNotification folder.

- Delete a record from the Employee table. You will receive a notification message resembling the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Delete</Info>
  <Source>Data</Source>
```

```
<Type>Change</Type>
</Notification>
```

Note that the value in the <Info> tag is "Delete". Hence, this message is received in C:\TestLocation\OtherNotification folder.

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

Performing Complex Operations After Receiving Notification Messages

For simplicity and better understanding, the orchestration in this topic copies messages to different folders based on the notification type. However, in real-world scenarios you might want to perform more complex operations. You can perform similar procedures as provided in this topic and build on them to perform the operations you wish. For example, you can change the orchestration to insert records in another table if you get a notification message for an Insert operation on the Employee table. In such a case, you can make appropriate changes within the Decide shape.

One such scenario is explained in detail in [Tutorial 2: Employee - Purchase Order Process](#).

See Also

Other Resources

[Receiving Query Notifications by Using BizTalk Server](#)

Receiving Query Notifications Incrementally Using BizTalk Server

Important

For the sake of brevity, this topic only describes how to receive notifications incrementally. In business scenarios, the orchestration must ideally include the logic to extract the kind of notification message received and then perform any subsequent operations. In other words, the orchestration described in this topic must be built on top of the orchestration described in [Processing Notification Messages to Perform Specific Tasks](#).

This topic demonstrates how to configure the SQL adapter to receive incremental query notification messages from a SQL Server database. To demonstrate incremental notifications, consider a table, *Employee*, with a "Status" column. When a new record is inserted to this table, the value of the Status column is set to 0. You can configure the adapter to receive incremental notifications by doing the following:

- Register for notifications using a SQL statement that retrieves all records that have Status column as 0. You can do so by specifying the SQL statement for the **NotificationStatement** binding property.
- For rows for which notification messages have been received, update the Status column to 1.

This topic demonstrates how to create a BizTalk orchestration and configure a BizTalk application to achieve this.

Configuring Notifications with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure receiving notifications from SQL Server. You must specify these binding properties while configuring the receive port in the BizTalk Server Administration console.

Note

You may choose to specify these binding properties when generating the schema for the **Notification** operation, even though it is not mandatory. If you do so, the port binding file that the Consume Adapter Service Add-in generates as part of the metadata generation also contains the values you specify for the binding properties. You can later import this binding file in the BizTalk Server Administration console to create the WCF-custom or WCF-SQL receive port with the binding properties already set. For more information about creating a port using the binding file, see [Configuring a Physical Port Binding Using a Port Binding File](#).

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for query notifications. The adapter gets a notification message from SQL Server only when the result set for the specified SQL statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to receive notifications from SQL Server, read further.

How This Topic Demonstrates Receiving Notification Messages

To demonstrate how the SQL adapter supports receiving notification messages from SQL Server, this topic will demonstrate how to configure the adapter to receive notifications for changes to an *Employee* table. Assume that the *Employee* table has columns *Employee_ID*, *Name*, and *Status*. Whenever a new employee is added, the value of the *Status* column is set to 0.

To demonstrate receiving notifications, do the following:

- Generate schema for the **Notification** (inbound operation), and **Select** (outbound operation) on the Employee table.
- Create an orchestration that has the following:
 - A receive location to receive notification messages. You can configure for notification by specifying the SELECT statement as:

```
SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0
```

 **Note**

You must specifically specify the column names in the statement as shown in this SELECT statement. Also, you must always specify the table name along with the schema name. For example, `dbo.Employee`.

- A send port to update the rows for which notification has already been sent. You do so by setting the value in the Status column to 1. You can do this as part of the Select operation by sending the following message to the adapter.

```
<Select xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <Columns>Employee_ID,Name,Status</Columns>
  <Query>where Status=0;UPDATE Employee SET Status=1 WHERE Status=0</Query>
</Select>
```

In this message, as part of the `<Query>` element, you specify the UPDATE statement to update the Status column. Note that this operation must be executed after receiving the notification messages so that the processed rows are updated. To do away with the overhead of waiting to get the notification response and then manually dropping a request message to update the rows, you will generate the request message for updating the rows within the orchestration itself. You can do so by using the **Construct Message** shape within an orchestration.

How to Receive Notification Messages from the SQL Server Database

Performing an operation on the SQL Server database using SQL adapter with BizTalk Server involves the procedural tasks described in [Building Blocks for Developing BizTalk Applications](#). To configure the adapter to receive notification messages, these tasks are:

1. Create a BizTalk project, and then generate schema for the **Notification** (inbound operation) and **Select** (outbound operation) on the Employee table. Optionally, you can specify values for the **InboundOperationType** and **NotificationStatement** binding properties.
2. Create a message in the BizTalk project for receiving notification from the SQL Server database.
3. Create messages in the BizTalk project for performing the Select information on the SQL Server database and receiving response messages.
4. Create an orchestration that does the following:
 - Receives notification message from SQL Server.
 - Creates a message to select and update the rows for which notification is received.
 - Sends this message to the SQL Server to update the rows and receives a response.
5. Build and deploy the BizTalk project.

6. Configure the BizTalk application by creating physical send and receive ports.

Note

For inbound operations, like receiving notification messages, you must only configure a one-way WCF-Custom or WCF-SQL receive port. Two-way WCF-Custom or WCF-SQL receive ports are not supported for inbound operations.

7. Start the BizTalk application.

This topic provides instructions to perform these tasks.

Sample Based on This Topic

A sample, IncrementalNotification, based on this topic is provided with the BizTalk Adapter Pack. For more information, see [Samples](#).

Generating Schema

You must generate the schema for the **Notification** operation and **Select** operation on Employee table. See [Retrieving Metadata for SQL Server Operations in Visual Studio](#) for more information about how to generate the schema. Perform the following tasks when generating the schema. Skip the first step if you do not want to specify the binding properties at design time.

1. Specify a value for **InboundOperationType** and **NotificationStatement** binding properties while generating the schema. For more information about this binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For instructions on how to specify binding properties, see [Specifying Binding Properties](#).
2. Select the contract type as **Service (Inbound operations)**.
3. Generate schema for the **Notification** operation.
4. Select the contract type as **Client (Outbound operations)**.
5. Generate schema for the **Select** operation on **Employee** table.

Defining Messages and Message Types

The schema that you generated earlier describes the "types" required for the messages in the orchestration. A message is typically a variable, the type for which is defined by the corresponding schema. Once the schema is generated, you must link it to the messages from the Orchestration view of the BizTalk project.

For this topic, you must create three messages—one to receive notifications from the SQL Server database, one to perform the Select operation, and one to receive the response for Select operation.

Perform the following steps to create messages and link them to schema.

To create messages and link to schema

1. Add an orchestration to the BizTalk project. From the Solution Explorer, right-click the BizTalk project name, point to **Add**, and then click **New Item**. Type a name for the BizTalk orchestration and then click **Add**.
2. Open the orchestration view window of the BizTalk project, if it is not already open. Click **View**, point to **Other Windows**, and then click **Orchestration View**.
3. In the **Orchestration View**, right-click **Messages**, and then click **New Message**.
4. Right-click the newly created message, and then select **Properties Window**.
5. In the **Properties** pane for **Message_1**, do the following:

Use this	To do this
s	

Identifier	Type NotifyReceive .
Message Type	From the drop-down list, expand Schemas , and select <i>SQLNotify.Notification</i> , where <i>SQLNotify</i> is the name of your BizTalk project. <i>Notification</i> is the schema generated for the Notification operation.

6. Repeat step 3 to create two new messages. In the **Properties** pane for the new message, do the following:

Set Identifier to	Set Message Type to
Select	<i>SQLNotify.TableOperation_dbo_Employee.Select</i> , where <i>TableOperation_dbo_Employee</i> is the schema generated for the Select operation
SelectResponse	<i>SQLNotify.TableOperation_dbo_Employee.SelectResponse</i>

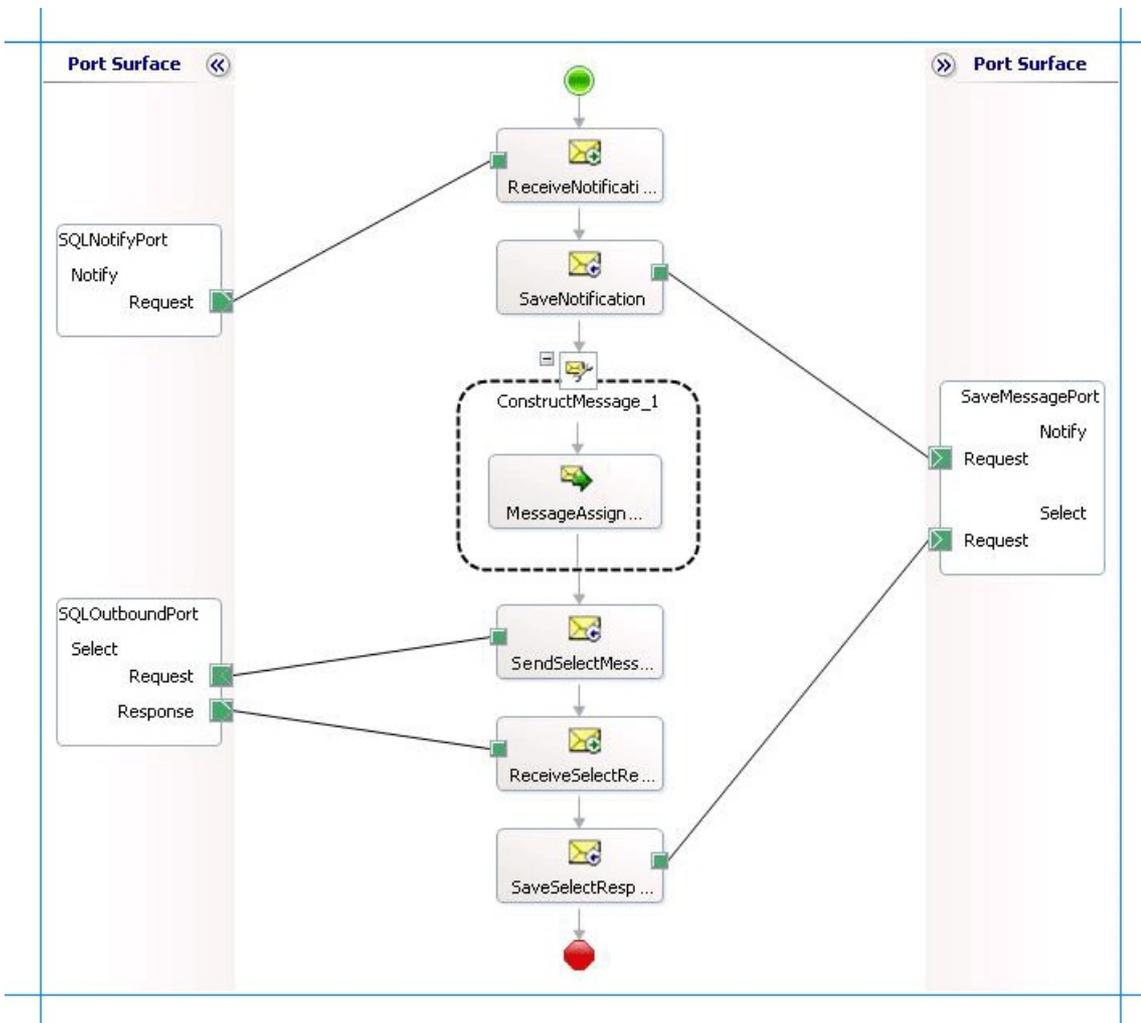
Setting up the Orchestration

You must create a BizTalk orchestration to use BizTalk Server for receiving notification messages from the SQL Server database and then updating the rows for which notification was received. In this orchestration, the adapter receives the notification message based on the SELECT statement specified for the **NotificationStatement** binding property. The notification message is received at a FILE location. Once the response is received, the orchestration constructs a message that will be used to update the rows for which notification is received. The response for this message is also received at the same FILE location.

So, your orchestration must contain the following:

- A one-way receive port to receive notification messages.
- A two-way send port to send messages to update rows and receive response for the same.
- A **Construct Message** shape to construct messages, to execute the Update operation, within the orchestration.
- A FILE send port to save the response for the Update operation.
- Receive and send shapes.

A sample orchestration resembles the following.



Adding Message Shapes

Make sure you specify the following properties for each of the message shapes. The names listed in the Shape column are the names of the message shapes as displayed in the just-mentioned orchestration.

Shape	Shape Type	Properties
ReceiveNotification	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveNotification</i> Set Activate to <i>True</i>
SaveNotification	Send	<ul style="list-style-type: none"> Set Name to <i>SaveNotification</i>
SendSelectMessage	Send	<ul style="list-style-type: none"> Set Name to <i>SendSelectMessage</i>
ReceiveSelectResponse	Receive	<ul style="list-style-type: none"> Set Name to <i>ReceiveSelectResponse</i>
SaveSelectResponse	Send	<ul style="list-style-type: none"> Set Name to <i>SaveSelectResponse</i>

Adding Construct Message Shape

You can use the **Construct Message** shape to generate a request message within the operation to perform the Select operation. To do so, you must add a **Construct Message** shape and within that a **Message Assignment** shape to your orchestration. For this example, the **Message Assignment** shape invokes code that generates a message that is sent to SQL Server to perform the Select operation. The **Message Assignment** shape also sets the action for the message to be sent to SQL Server.

For the construct message shape, set the **Message Constructed** property to **Select**.

The code to generate the response could be part of the same Visual Studio solution as your BizTalk project. A sample code for generating a response message looks like this.

```

namespace SampleMessageCreator
{
    public class SampleMessageCreator
    {
        private static XmlDocument Message;
        private static string XmlFileLocation;
        private static string ResponseDoc;
        public static XmlDocument XMLMessageCreator()
        {
            XmlFileLocation = "C:\\\\TestLocation\\CreateMessage";
            try
            {
                ResponseDoc = (Directory.GetFiles(XmlFileLocation, "*.xml", SearchOption.To
                pDirectoryOnly))[0];
            }
            catch (Exception ex)
            {
                Console.WriteLine("Trying to get XML from: " + XmlFileLocation);
                Console.WriteLine("EXCEPTION: " + ex.ToString());
                throw ex;
            }
            //Create Message From XML
            Message = new XmlDocument();
            Message.PreserveWhitespace = true;
            Message.Load(ResponseDoc);
            return Message;
        }
    }
}

```

For the preceding code excerpt to be able to generate a request message, you must have an XML request message (for the Select operation on the Employee table) in the location specified for the `XmlFileLocation` variable.

Note

After you build the project, `SampleMessageCreator.dll` will be created in the project directory. You must add this DLL to the global assembly cache (GAC). Also, you must add the `SampleMessageCreator.dll` as a reference in the BizTalk project.

Add the following expression to invoke this code from the **Message Assignment** shape and to set the action for message. To add an expression, double-click the **Message Assignment** shape to open the Expression Editor.

```

Select = SampleMessageCreator.SampleMessageCreator.XMLMessageCreator();
Select(WCF.Action) = "TableOp/Select/dbo/Employee";

```

Adding Ports

Make sure you specify the following properties for each of the logical ports. The names listed in the Port column are the names of the ports as displayed in the orchestration.

Port	Properties
SQLNotifyPort	<ul style="list-style-type: none"> • Set Identifier to <i>SQLNotifyPort</i> • Set Type to <i>SQLNotifyPortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Receive</i>

SaveMessagePort	<ul style="list-style-type: none"> • Set Identifier to <i>SaveMessagePort</i> • Set Type to <i>SaveMessagePortType</i> • Set Communication Pattern to <i>One-Way</i> • Set Communication Direction to <i>Send</i> • Create an operation <i>Notify</i>. This operation is used for notification messages. • Create an operation <i>Select</i>. This operation is used for select response messages.
SQLOutboundPort	<ul style="list-style-type: none"> • Set Identifier to <i>SQLOutboundPort</i> • Set Type to <i>SQLOutboundPortType</i> • Set Communication Pattern to <i>Request-Response</i> • Set Communication Direction to <i>Send-Receive</i>

Specify Messages for Action Shapes and Connect to Ports

The following table specifies the properties and their values that you should set to specify messages for action shapes and to link the messages to the ports. The names listed in the Shape column are the names of the message shapes as displayed in the orchestration mentioned earlier.

Shape	Properties
ReceiveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>SQLNotifyPort.Notify.Request</i>
SaveNotification	<ul style="list-style-type: none"> • Set Message to <i>NotifyReceive</i> • Set Operation to <i>SaveMessagePort.Notify.Request</i>
SendSelectMessage	<ul style="list-style-type: none"> • Set Message to <i>Select</i> • Set Operation to <i>SQLOutboundPort.Select.Request</i>
ReceiveSelectResponse	<ul style="list-style-type: none"> • Set Message to <i>SelectResponse</i> • Set Operation to <i>SQLOutboundPort.Select.Response</i>
SaveSelectResponse	<ul style="list-style-type: none"> • Set Message to <i>SelectResponse</i> • Set Operation to <i>SaveMessagePort.Select.Request</i>

After you have specified these properties, the message shapes and ports are connected and your orchestration is complete.

You must now build the BizTalk solution and deploy it to a BizTalk Server. For more information, see "Building and Running Orchestrations" at <http://go.microsoft.com/fwlink/?LinkId=102359>.

Configuring the BizTalk Application

After you have deployed the BizTalk project, the orchestration you created earlier is listed under the **Orchestrations** pane in the BizTalk Server Administration console. You must use the BizTalk Server Administration console to configure the application. For more information about configuring an application, see <http://go.microsoft.com/fwlink/?LinkID=196961>.

Configuring an application involves:

- Selecting a host for the application.
- Mapping the ports that you created in your orchestration to physical ports in the BizTalk Server Administration console. For this orchestration you must:
 - Define a physical WCF-Custom or WCF-SQL one-way receive port. This port listens for notifications coming from the SQL Server database. For information about how to create ports, see [Manually Configuring a Physical Port Binding to the SQL Adapter](#). Make sure you specify the following binding properties for the receive port.

<p>◆ Important</p> <p>You do not need to perform this step if you specified the binding properties at design time. In such a case, you can create a WCF-custom or WCF-SQL receive port, with the required binding properties set, by importing the binding file created by the Consume Adapter Service Add-in. For more information see Configuring a Physical Port Binding Using a Port Binding File.</p>

Binding Property	Value	
InboundOperationType	Set this to Notification .	
NotificationStatement	Set this to: <table border="1" style="margin-left: 20px;"> <tr> <td> <pre>SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0</pre> </td> </tr> </table> <p> Note You must specifically specify the column names in the statement as shown in this SELECT statement. Also, you must always specify the table name along with the schema name. For example, <code>dbo.Employee</code>.</p>	<pre>SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0</pre>
<pre>SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0</pre>		
NotifyOnListenerStart	Set this to True .	

For more information about the different binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

<p> Note</p> <p>We recommend configuring the transaction isolation level and the transaction timeout while performing inbound operations using the SQL adapter. You can do so by adding the service behavior while configuring the WCF-Custom or WCF-SQL receive port. For instruction on how to add the service behavior, see Configure Transaction Isolation Level and Transaction Timeout.</p>

- Define a physical WCF-Custom or WCF-SQL send port to send messages to the SQL Server database. You must also specify the action in the send port.
- Define a location on the hard disk and a corresponding file port where the BizTalk orchestration will drop the messages from the SQL Server database. These will be the notification messages received from SQL Server and messages for the Select and Update operation you perform through the WCF-Custom or WCF-SQL send port.

Starting the Application

You must start the BizTalk application for receiving notification messages from the SQL Server database and for performing the subsequent Select and Update operations. For instructions on starting a BizTalk application, see "How to Start an Orchestration" at <http://go.microsoft.com/fwlink/?LinkId=102387>.

At this stage, make sure:

- The WCF-Custom or WCF-SQL one-way receive port, which receives the notification messages from the SQL Server database is running.
- The WCF-Custom or WCF-SQL send port to perform Select and Update operations on the Employee table is running.
- The FILE send port, which receives messages from SQL Server, is running.
- The BizTalk orchestration for the operation is running.

Executing the Operation

To execute this operation, you must insert a record into the Employee table. Let us assume you insert a record with following details:

```
Name = John Smith
Designation = Manager
Salary = 100000
```

Also, make sure the XML message to perform Select and Update operations is available at C:\TestLocation\MessageIn. The XML file should resemble the following:

```
<Select xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <Columns>Employee_ID,Name,Status</Columns>
  <Query>where Status=0;UPDATE Employee SET Status=1 WHERE Status=0</Query>
</Select>
```

Once the record is inserted, the following set of actions take place, in the same sequence:

- The adapter receives a notification message that resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Insert</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

This message notifies that a record was inserted in the Employee table. Note that the value for the <Info> element is "Insert".

- The adapter executes the Select operation. Because the Select operation XML also includes an Update statement, the Update statement is also executed. The next response from SQL Server is for the Select statement.

```
<?xml version="1.0" encoding="utf-8" ?>
<SelectResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <SelectResult>
    <Employee xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/Tables/dbo">
      <Employee_ID>10006</Employee_ID>
```

```
<Name>John</Name>
<Status>0</Status>
</Employee>
</SelectResult>
</SelectResponse>
```

This response shows that a record was inserted into the Employee table and the Status for that record is 0.

- As part of the Select statement, the Update statement is also executed and the Status column for the new record is changed to 1. This again triggers another notification from SQL Server and the adapter receives a corresponding notification message, that resembles the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification/">
  <Info>Update</Info>
  <Source>Data</Source>
  <Type>Change</Type>
</Notification>
```

This message notifies that a record was updated in the Employee table. Note that the value for the <Info> element is "Update".

- After the second notification, the adapter executes the Select statement. However, because there are no records now that have Status as 0, the adapter gets an empty response resembling the following.

```
<?xml version="1.0" encoding="utf-8" ?>
<SelectResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <SelectResult />
</SelectResponse>
```

Best Practices

After you have deployed and configured the BizTalk project, you can export configuration settings to an XML file called the binding file. Once you generate a binding file, you can import the configuration settings from the file, so that you do not need to create the send ports and receive ports for the same orchestration. For more information about binding files, see [Reusing Adapter Bindings](#).

See Also

Other Resources

[Receiving Query Notifications by Using BizTalk Server](#)

Receiving Query Notifications On Multiple Receive Locations

Consider a scenario where you have multiple receive locations created as part of different BizTalk applications configured to receive query notifications for the same table (e.g. Employee) in the same database. If a hundred records are inserted into the same table, all the receive locations will get the notification message. To effectively receive notifications across multiple receive locations, you can call operations from your BizTalk application in such a way that if a notification is received by one receive location, the other receive location does not get the same notification. So, you can effectively load-balance notifications received on multiple locations.

The tasks required to set up an orchestration to load-balance receiving notifications are same as that for [Receiving Query Notifications Incrementally Using BizTalk Server](#). This topic lists the only the difference between the two approaches.

Load-Balancing Query Notifications Across Multiple Receive Locations

Like in the topic [Receiving Query Notifications Incrementally Using BizTalk Server](#), you configured incremental notifications by executing an UPDATE statement on the records that are already notified for. To configure load-balancing, you could execute a stored procedure that deletes the records that have been notified for. For example, consider a stored procedure PROCESS_EMPLOYEE with the following definition:

```
DECLARE @var int
SELECT TOP 1 @var = Employee_ID FROM Employee
SELECT * FROM Employee WHERE Employee_ID=@var
DELETE FROM Employee WHERE Employee_ID=@var
```

When you execute this stored procedure as part of the BizTalk application, the record for which notification is already received gets deleted. So, the other receive location gets notification for the next record.

Here are the high-level steps you must perform to configure load-balancing for receiving notifications.

1. Create schema for **Notification** (inbound operation) and **PROCESS_EMPLOYEE** stored procedure (outbound operation).
2. Add an orchestration and add three messages for receiving notification, executing stored procedure, and getting response for the stored procedure.
3. Create an orchestration by adding Send and Receive shapes, Construct Message shape, and ports. You can use the same sample code for constructing a message to invoke the PROCESS_EMPLOYEE stored procedure. Note that while performing the operation in BizTalk Server Administration console, you must have the request message for the PROCESS_EMPLOYEE stored procedure in the location C:\TestLocation\MessageIn. You do so because the code snippet you invoke as part of the orchestration created in [Receiving Query Notifications Incrementally Using BizTalk Server](#) creates a request message based on the request XML present in C:\TestLocation\MessageIn.
4. Build and deploy the application. To demonstrate load-balancing, you must deploy this orchestration at least on two different computers that have BizTalk Server and SQL adapter installed.
5. In the BizTalk Server Administration console on both the computers, specify the following binding properties for the WCF-Custom or WCF-SQL receive location:

Binding Property	Value
InboundOperationType	Set this to Notification .

NotificationStatement	Set this to: <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0 </div> <p>  Note For notification statements, you must always specify the table name along with the schema name. For example, <code>dbo.Employee</code>. </p>
NotifyOnListenerStart	Set this to True .

6. Start the BizTalk application.
7. To start receiving notifications, insert a hundred records into the EMPLOYEE table. While doing so, make sure the request XML for invoking the PROCESS_EMPLOYEE stored procedure is available in C:\TestLocation\MessageIn.
8. Monitor the location (on both the computers) where the BizTalk application will be dropping the notification messages. You will notice that of the hundred records inserted, one location gets notifications for some records while the other location gets notification for the remaining records. Together, both the locations will get notification for all the hundred records.

See Also

Other Resources

[Receiving Query Notifications by Using BizTalk Server](#)

Receiving Query Notifications After a Receive Location Breakdown

Consider a scenario where you have a BizTalk application that receives database change notification messages when changes are made to the EMPLOYEE table. If the receive location configured as part of the BizTalk application breaks down, and simultaneously records are added into the EMPLOYEE table, you will not receive notifications for the recently added records. You will also not know when the receive location is available again. The SQL adapter exposes a binding property, **NotifyOnListenerStart**, that you can configure to get a notification that the receive location has recovered. You can specify the following values for the **NotifyOnListenerStart** binding property:

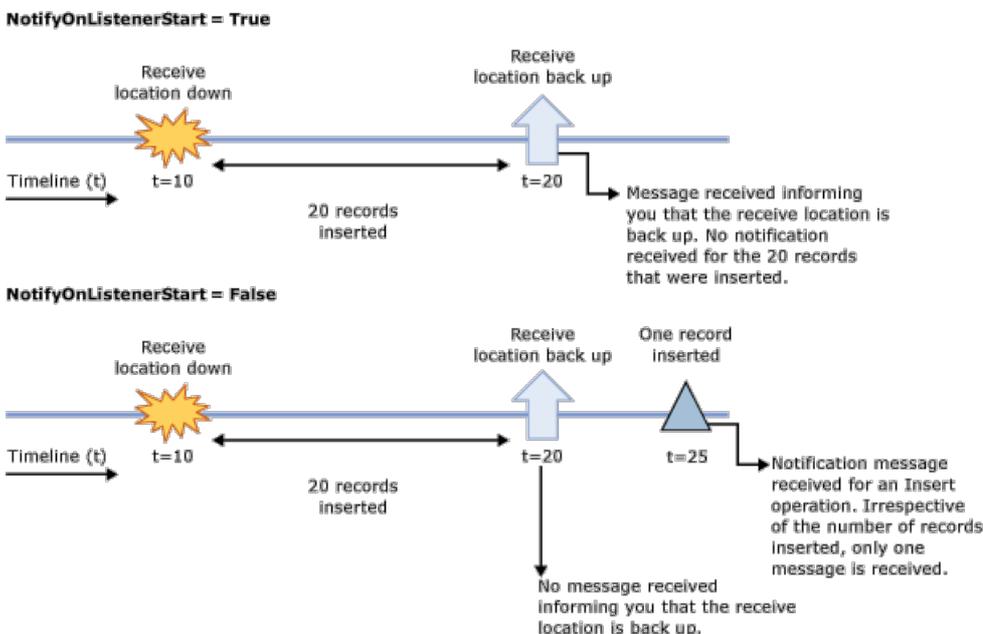
- Set this property to **True**, to receive a notification informing that the receive location is available, as soon as the receive location recovers.
- Set this property to **False**, to not receive a notification informing that the receive location has recovered, after the receive location recovers.

Default is **True**.

Configuring the SQL Adapter Behavior

For either of the approaches, you do not need to perform any specific tasks while generating metadata or while configuring the BizTalk application. You only need to set the **NotifyOnListenerStart** binding property accordingly on the WCF-Custom or WCF-SQL receive location. To create the BizTalk application, you must perform the same set of tasks as described in [Receiving Query Notifications Incrementally Using BizTalk Server](#). However, when configuring the BizTalk application using BizTalk Server, you can try changing the value of the **NotifyOnListenerStart** binding property and see the difference in the two configurations.

The following figure demonstrates how the notifications are received based on the value of the **NotifyOnListenerStart** binding property.



Note that in the first scenario, when the **NotifyOnListenerStart** is set to **true** and records are inserted into the database table while the receive location was down, the adapter only sends you a notification message when the receive location comes back up. The adapter does not perform any operation to process the records that were inserted while the receive location was down. The adapter client must implement the relevant logic in their application to process the records that were inserted while the receive location was down.

See Also

Other Resources

[Receiving Query Notifications by Using BizTalk Server](#)

Developing Applications by Using the WCF Service Model

Windows Communication Foundation (WCF) provides a programming model called the WCF service model, as an alternative to the WCF channel programming model.

The WCF service model uses familiar .NET paradigms to hide the complexities of exchanging SOAP messages over a channel. The service model accomplishes this simplification by reading the entire SOAP message into memory before copying the information into .NET data structures. Loading long messages into memory, however, may not be practical for some applications. In these cases, developers should use the WCF channel model. For more information about using the WCF channel model, see [Developing Applications by Using the WCF Channel Model](#).

At the lowest level, WCF presents the WCF channel model, in which clients invoke operations on a service by exchanging SOAP messages over a channel established between client and service endpoints. The WCF channel model exposes data types and methods that enable you to operate directly on the WCF channel architecture. The WCF channel model provides you with direct control over the contents of the SOAP messages you create and over the way both your application and the Microsoft BizTalk Adapter for SQL Server consume them. However, creating well-formed SOAP messages to send over a channel and validating the reply messages returned can be a detailed and exacting task.

The WCF service model uses proxy classes to invoke operations on a target service or to receive operations from a client. The SQL adapter exposes the SQL Server database as a WCF service on which you can invoke operations.

- The proxy class that is used to invoke operations on a target service is called a WCF client class. This class models the operations exposed by a service as .NET methods with strongly-typed parameters. By using the WCF service model, you can invoke the operations exposed by the SQL adapter as .NET methods on the WCF client. For more information about WCF clients, see "WCF Client Overview" at <http://go.microsoft.com/fwlink/?LinkId=91458>.

You can use either of the following tools to generate a WCF client class and associated helper code from the service metadata that the SQL adapter exposes:

- **The ServiceModel Metadata Utility Tool (svcutil.exe)**, which ships with WCF.
- **The Add Adapter Service Reference Visual Studio Plug-in**, which ships with SQL adapter and is integrated with the Visual Studio design experience. This tool presents a standard Microsoft Windows interface that provides powerful browsing and searching capabilities on operations that the adapter exposes. For more information about how to generate a WCF client application, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

The topics in this section contain information, procedures, and examples to help you create and use the WCF service model to develop applications by using the SQL adapter.

In This Section

- [Overview of Using the WCF Service Model with the Adapter](#)
- [Metadata and the WCF Service Model](#)
- [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#)
- [Specifying a Client Binding for the SQL Adapter](#)
- [Performing Insert, Update, Delete, and Select Operations by Using the WCF Service Model](#)
- [Performing Operations on Tables and Views with Large Data Types by Using the WCF Service Model](#)
- [Invoking Stored Procedures Using the WCF Service Model](#)
- [Invoking Scalar Functions in SQL Server by Using the WCF Service Model](#)
- [Invoking Table-Valued Functions in SQL Server by Using the WCF Service Model](#)

- [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using WCF Service Model](#)
- [Polling SQL Server by Using the SQL Adapter with WCF Service Model](#)
- [Receiving Query Notifications Using the WCF Service Model](#)

See Also

Other Resources

[Development](#)

Overview of Using the WCF Service Model with the Adapter

The Microsoft BizTalk Adapter for SQL Server exposes a SQL Server operation as a WCF service. To perform operations on SQL Server artifacts, for example to invoke a stored procedure, you invoke an operation on the adapter, which, in turn, performs the operation on the SQL Server. Your code therefore acts as a client to the WCF service presented by the adapter.

In the Windows Communication Foundation (WCF) service model, the service contract that exists between a client and a service is represented as a .NET interface, and operations are represented as methods on this interface. The SQL adapter and WCF provide tools that enable you to generate this interface for targeted operations from the metadata that the adapter exposes. These tools also create a WCF client class that can be used to invoke the operations exposed in the service interface. A client application can call the methods of the WCF client class to invoke operations on the adapter. To implement a service to receive inbound operations from the SQL adapter, you implement the interface generated for the inbound operations.

The following section explains how to use the WCF service model to invoke operations with a WCF client.

Invoking Operations on the SQL Server with a WCF Client

To use the WCF service model to invoke operations on the SQL adapter, you must first generate a WCF client class for the target operations. You can then create an instance of this class, a WCF client, and call its methods to perform these operations on the SQL Server system. This section provides an outline of how a typical .NET adapter client application looks like. Detailed explanations on how to perform different operations on the SQL Server database using the adapter are provided in specific topics.

To invoke operations on the SQL adapter

1. Generate a WCF client class and helper code. Use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class targeted at the SQL Server database artifacts with which you want to work. For more information about how to generate a WCF client, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).
2. Create a WCF client instance and configure the WCF client. Configuring the WCF client involves specifying the binding and endpoint address (connection URI) that the client will use. You can do this either imperatively in code or declaratively in configuration. The following code creates a WCF client that targets the **Select** operation on the **Employee** table in a SQL Server database. It also sets the credentials for the SQL Server database. The WCF client is initialized from configuration.

```
TableOp_dbo_EmployeeClient client = new TableOp_dbo_EmployeeClient("SqlAdapterBinding_
TableOp_dbo_Employee"); //picking the binding and address from the app.config

client.ClientCredentials.UserName.UserName = "myuser";
client.ClientCredentials.UserName.Password = "mypassword";
```

Note

You can either specify the client binding and endpoint address in the code or declare it in the app.config configuration file. The preceding code snippet uses the latter. For more information on how to use either approaches, see [Specifying a Client Binding for the SQL Adapter](#).

3. Open the WCF client.

```
client.Open();
```

4. Invoke methods on the WCF client created in preceding step to perform a Select operation on the SQL server database. The following code invokes the Select method of the WCF client to invoke the SELECT statement on a SQL Server database table.

```
client.Select("...", "where [Name] = 'John Smith'");
```

5. Close the WCF client.

```
client.Close();
```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Metadata and the WCF Service Model

In the WCF service model, you use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool (svcutile.exe) to do the following:

- Generate a service contract—the WCF service contract—through which your code can receive operations from the adapter. This .NET interface represents the service contract for target operations.
- Generate proxy classes—the WCF client class—through which your code can invoke operations on the adapter.
- Annotated classes that represent the supporting message contracts, operation contracts, and data contracts for the service contract.

For help in understanding the structure of this generated code, see "Understanding Generated Client Code" at <http://go.microsoft.com/fwlink/?LinkId=98365>. This topic specifically describes code that svcutil.exe generates, but its content is also applicable to the code that the Add Adapter Service Reference Plug-in generates.

For information about how to generate a WCF client class or WCF service contract for target operations and about the differences between svcutil.exe and the Add Adapter Service Reference Plug-in, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Generating a WCF Client or WCF Service Contract for SQL Server Artifacts

You can use the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF client class targeted at selected operations on SQL Server artifacts. You can also use the ServiceModel Metadata Utility Tool (svcutil.exe) to generate the WCF client class; however, the Add Adapter Service Reference Plug-in exposes the functionality of the ServiceModel Metadata Utility Tool through a standard Microsoft Windows interface. It also provides browse and search capabilities that are not available with the svcutil.exe tool, and generates a configuration file based on the binding properties that you select when you connect to the SQL Server database.

Generating a WCF Client Class by Using the Add Adapter Service Reference Plug-in

Perform the following steps to generate a WCF client class by using the Add Adapter Service Reference Plug-in.

To generate a WCF client class

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Retrieving Metadata for SQL Server Operations in Visual Studio](#) to connect to SQL Server and to browse and search for operations. To create a WCF client class for the operations that you select, be sure that **Client (Outbound operations)** is selected from the **Select contract type** drop-down list. (This is the default).
3. After you select all of the operations that you want to target, click **OK** to generate the WCF client class.

The Add Adapter Service Reference Plug-in adds two files to your project:

- **The WCF client code file.** This file contains the generated WCF client class and helper code for the operations that you selected. The first time you run the Add Adapter Service Reference Visual Studio Plug-in, it will generate this file with the default name **SQLAdapterBindingClient.cs**. If you run it again, the next file it generates will be called **SQLAdapterBindingClient1.cs**. The number suffix will increase by 1 for every new file you generate. You can also change the default prefix **SQLBinding** by entering a different prefix in the **Filename prefix** field of the Add Adapter Service Reference Visual Studio Plug-in before selecting **OK** to generate the file.
- **App.config.** This file contains a binding configuration and client endpoint configurations that are based on the selections you made when you configured the connection for the Add Adapter Service Reference Plug-in.

◆ Important

While using the Add Adapter Service Reference Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Generating a WCF Service Contract by Using the Add Adapter Service Reference Plug-in

For inbound operations such as polling the SQL Server database or receiving notifications from the database, the SQL adapter either executes a query specified by the client application (in case of polling) or registers a query with SQL Server (in case of notification). In both the scenarios, the adapter sends the inbound message from SQL Server database to the consuming. In such a case, the consuming application acts as a service and the SQL adapter acts as the client. You must, therefore, implement a WCF service that can receive inbound operations from the adapter. To do this, you use the Add Adapter Service Reference Plug-in to generate a .NET interface that represents the service contract that is surfaced by the adapter for the inbound operations. This .NET interface is also called a WCF service contract. You then implement this interface to create the WCF service that you can use to receive the inbound operations.

Perform the following steps to generate a WCF service contract by using the Add Adapter Service Reference Plug-in.

To generate a WCF service contract for inbound operations

1. In Visual Studio Solution Explorer, right-click your project, and then click **Add Adapter Service Reference**.
2. After the **Add Adapter Service Reference** dialog box opens, follow the steps in [Connecting to SQL Server in Visual Studio Using Add Adapter Service Reference Plug-in](#) to connect to the SQL Server

database.

Important

If you are generating WCF service contract for **TypedPolling** inbound operation, you must specify the **InboundID** as part of the connection URI and **PollingStatement** binding property.

3. After you have connected to the SQL Server database, select **Service (Inbound operations)** from the **Select contract type** drop-down list.
4. In the **Select a category** box, click the root node (*/*), select the inbound operation from the **Available categories and operations** box, and then click **Add**.
5. To generate the WCF service contract for the inbound operation, click **OK**.

The Add Adapter Service Reference Plug-in adds three files to your project:

- **SqlAdapterBindingInterface.cs**. This file contains the generated WCF service contract (interface) and helper code for the inbound operation.
- **SqlAdapterBindingService.cs**. This file contains a class that implements the interface defined in `SqlAdapterBindingInterface.cs`. You can implement the business logic that processes the records returned by the inbound operation.
- **app.config**. This file contains a binding configuration, endpoint behaviors, and service endpoint configuration that are based on the selections you made when you configured the binding and connection for the Add Adapter Service Reference Plug-in.

Important

While using the Add Adapter Service Reference Plug-in, if you do not specify a value for a binding property of type string and whose default value is null then that binding property will not be available in the app.config file. You must manually add the binding property and its value in the app.config file, if required.

Generating a WCF Client Class by Using svcutil.exe

You can use svcutil.exe to generate a WCF client class for your application. You must configure svcutil.exe to use it with the SQL adapter.

Svcutil.exe generates the WCF client class in an output file with a default file name of **output.cs**. You must manually include this file in your Visual Studio project. For more information about svcutil.exe, see <http://go.microsoft.com/fwlink/?LinkId=139432>.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Specifying a Client Binding for the SQL Adapter

After you have generated the WCF client class, you can create a WCF client (instance) and invoke its methods to consume the Microsoft BizTalk Adapter for SQL Server. For information about how to generate the WCF client class and helper code for operations that the SQL adapter exposes, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

To create the WCF client, you must specify an endpoint address and a binding. The endpoint address must contain a valid SQL connection URI, and the binding must be an instance of a SQL Binding (**sqlBinding**). For more information about the SQL connection URI, see [Specifying the Connection URI](#). You must specify the user credentials as part of the connection URI. You may use the **ClientCredentials** property of the WCF client, as explained in this topic.

You can specify the SQL binding and the endpoint address in your code or in a configuration file. When you use the Add Adapter Service Reference Visual Studio Plug-in to generate the WCF client class, a configuration file (app.config) is also created for your project. This file contains configuration settings that reflect the binding properties and connection information (except credentials) that you specified when you connected to the SQL database with the Add Adapter Service Reference Plug-in.

Specifying the Binding and Endpoint Address in Code

The following code shows how to create a WCF client by specifying the binding and endpoint address in code by using the **ClientCredentials** property of the WCF client.

```
SqlAdapterBinding binding = new SqlAdapterBinding();
EndpointAddress sqlAddress = new EndpointAddress("mssql://<sql_server_name>//<database_name>?");

TableOp_dbo_CustomerClient client = new TableOp_dbo_CustomerClient (binding, sqlAddress);

client.ClientCredentials.UserName.UserName = "USER";
client.ClientCredentials.UserName.Password = "PASSWORD";

client.Open();
```

Specifying the Binding and Endpoint Address in a Configuration File

The following code shows how to create a WCF client by specifying the binding and endpoint address in an app.config file.

```
TableOp_dbo_CustomerClient client = new TableOp_dbo_CustomerClient("SqlAdapterBinding_TableOp_dbo_Customer");

client.ClientCredentials.UserName.UserName = "USER";
client.ClientCredentials.UserName.Password = "PASSWORD";

client.Open();
```

The following XML shows the configuration file created for the Customer table by the Add Adapter Service Reference Plug-in. This file contains the client endpoint configuration referenced in the preceding example.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <sqlBinding>
        <binding name="SqlAdapterBinding" closeTimeout="00:01:00" openTimeout="00:01:00"
1:00"
          receiveTimeout="00:10:00" sendTimeout="00:01:00" maxConnectionPoolSize=
"100"
          encrypt="false" useAmbientTransaction="true" batchSize="20"
          polledDataAvailableStatement="" pollingStatement="" pollingIntervalInSe
conds="30"
          pollWhileDataFound="false" notificationStatement="" notifyOnListenerSta
rt="true"
          enableBizTalkCompatibilityMode="true" chunkSize="4194304"
```

```

        inboundOperationType="Polling" useDatabaseNameInXsdNamespace="false"
        allowIdentityInsert="false" enablePerformanceCounters="false"
        xmlStoredProcedureRootNodeName="" xmlStoredProcedureRootNodeNamespace=""
" />
    </sqlBinding>
</bindings>
<client>
    <endpoint address="mssql://<sql_server_name>//<database_name>?" binding="sqlBin
ding"
        bindingConfiguration="SqlAdapterBinding" contract="TableOp_dbo_Customer"
        name="SqlAdapterBinding_TableOp_dbo_Customer" />
    </client>
</system.serviceModel>
</configuration>

```

If a project has more than one WCF client, there will be multiple client endpoint entries defined in the configuration file. Each WCF client entry will have a unique name based on its binding configuration and target SQL Server artifact; for example, "SqlAdapterBinding_TableOp_dbo_Customer". If you connect multiple times to create the WCF clients in your project, multiple binding configuration entries will be created, one for each connection. These binding configuration entries will be named in the following manner: SqlAdapterBinding, SqlAdapterBinding1, and so on. Each client endpoint entry created during a specific connection will reference the binding entry created during that connection.

See Also

Tasks

[Specifying the Connection URI](#)

Concepts

[Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#)

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing Insert, Update, Delete, and Select Operations by Using the WCF Service Model

The Microsoft BizTalk Adapter for SQL Server discovers a set of basic Insert, Select, Update, and Delete operations on SQL Server database tables and views. By using these operations, you can perform simple SQL Insert, Select, Update, and Delete statements qualified by a Where clause on a target table or view. This topic provides instructions on how to perform these operations using the WCF service model.

For more information on how the adapter supports these operations, see [Performing Basic Insert, Update, Delete, and Select Operations on Tables and Views](#).

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

About the Examples Used in this Topic

The example in this topic performs operations on the Employee table. The Employee table is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **EmployeeBasicOps**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Client Class

The name of the WCF client generated for the basic SQL operations that the SQL adapter discovers is based on the name of the table or view, as listed in the following table.

SQL Server Database Artifact	WCF Client Name
Table	TableOp_[Schema]_[TABLE_NAME]Client
View	ViewOp_[Schema]_[VIEW_NAME]Client

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[TABLE_NAME] = The name of the table; for example, Employee.

[VIEW_NAME] = The name of the view; for example, Employee_View.

Method Signature for Invoking Operations on Tables

The following table shows the method signatures for the basic operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Insert	long[] Insert([TABLE_NS].[TABLE_NAME][] Rows);
Select	[TABLE_NS].[TABLE_NAME][] Select(string COLUMNS, string QUERY);
Update	int Update([TABLE_NS].[TABLE_NAME].RowPair[] Rows);
Delete	int Delete([TABLE_NS].[TABLE_NAME][] Rows);

[TABLE_NS] = The name of the table namespace; for example, schemas.microsoft.com.Sql_2008_05.Types.Tables.dbo.Employee.

[TABLE_NAME] = The name of the table; for example, Employee.

As an example, the following code shows the method signatures for a WCF client class generated for the Delete, Insert, Select and Update operations on the Employee table under the default "dbo" schema.

```
public partial class TableOp_dbo_EmployeeClient : System.ServiceModel.ClientBase<TableOp_db
```

```

o_Employee>, TableOp_dbo_Employee {
    public int Delete(schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] Rows)
    ;

    public long[] Insert(schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] Rows);

    public schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] Select(string Columns, string Query);

    public int Update(schemas.microsoft.com.Sql._2008._05.TableOp.dbo.Employee.RowPair[] Rows);
}

```

In this snippet, TableOp_dbo_EmployeeClient is the name of the WCF class in the SqlAdapterBindingClient.cs generated by the Add Adapter Service Reference Plug-in.

Parameters for Table Operations

This section provides the parameters required by each table operation

Insert Operation

Insert operation type	RECORDSET
Multiple record	A collection of INSERTRECORDS that should be inserted into the table.

The insert operation returns an array of Long data type and stores the identity values of the inserted rows, if any. If there is no identity column in a table, the return value is NULL.

Select Operation

COLUMN_NAMES	QUERY
A comma-delimited list of column names in the target; for example, "Employee_ID, Designation". The column list specifies the columns of the target that should be returned in the result set. Columns not specified in the column list will be set to their .NET default values in the returned record set. For nullable columns, this value is null .	The contents of a SQL WHERE clause that specifies the target rows of the query; for example, "Designation = 'Manager'". You can set this parameter to null to return all rows of the target.

The return value of the Select operation is a strongly-typed result set that contains the specified columns and rows from the target table or view.

Update Operation

First row of the pair	Second row of the pair
The first record of the record pair corresponds to new values that need to be updated, that is, it corresponds to the SET clause of the UPDATE statement. This can be set using <code>RowPair.After</code> .	The second record of the record pair corresponds to the old values of the rows, that is, it corresponds to the WHERE clause of the UPDATE statement. This can be set using <code>RowPair.Before</code> .

The return value of the Update operation is of Int32 data type, and denotes the number of rows updated.

◆ Important

While specifying the record that has to be updated, you must provide values for all the columns, even if all values are not being updated. For example, if a row has five columns and the Update operation updates only 2 columns, as part of RowPair.Before, you must pass all the 5 column values. However, for RowPair.After, you can specify only the columns that will be updated.

Delete Operation

The Delete operation takes as input a strongly-typed array of records. The return value of the Delete operation is of Int32 data type, and denotes the number of rows deleted.

Creating a WCF Client to Invoke Operations on Tables and Views

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke basic Insert, Select, Update, Delete operations on a table.

To create a WCF client to perform operations on tables

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the Insert, Select, Update, and Delete operation on the Employee table. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

◆ Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and create a client as described in the snippet below.

```
TableOp_dbo_EmployeeClient client = new TableOp_dbo_EmployeeClient("SqlAdapterBinding_TableOp_dbo_Employee");
client.ClientCredentials.UserName.UserName = "<Enter user name here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `TableOp_dbo_EmployeeClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_TableOp_dbo_Employee` is the name of the client endpoint configuration and is defined in the app.config. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

📌 Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the Insert operation on the Employee table.

```
long[] recordsInserted;

schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] insertRecord =
new schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[1];

insertRecord[0] = new schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee();
insertRecord[0].Name = "John Smith";
```

```

insertRecord[0].Designation = "Manager";
insertRecord[0].Salary = 500000;

try
{
    Console.WriteLine("Inserting new table entry...");
    recordsInserted = client.Insert(insertRecord);
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("Record inserted");
Console.WriteLine("Press any key to continue ...");
Console.ReadLine();

```

You can replace the preceding code snippet to perform Select, Update, or Delete operations as well. You can also append the code snippets to perform all operation in a single application. For code snippets on how to perform these operations, see [Select Operation](#), [Update Operation](#), and [Delete Operation](#) respectively.

7. Close the client as described in the snippet below:

```

client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();

```

8. Build the project and then run it. The application inserts a record in the Employee table.

Select Operation

The following code shows a Select operation that targets the Employee table. The Select operation selects the last record inserted into the table. The returned records are written to the console.

```

schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] selectRecords;

try
{
    Console.WriteLine("Selecting Row...");
    selectRecords = client.Select("select * from Employee where [Employee_ID] = (select IDENT_CURRENT('Employee'))");
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("The details of the newly added employee are:");
Console.WriteLine("*****");
for (int i = 0; i < selectRecords.Length; i++)
{
    Console.WriteLine("Employee ID      : " + selectRecords[i].Employee_ID);
    Console.WriteLine("Employee Name    : " + selectRecords[i].Name);
    Console.WriteLine("Employee Designation: " + selectRecords[i].Designation);
    Console.WriteLine();
}
Console.WriteLine("*****");
Console.WriteLine("Press any key to continue ...");
Console.ReadLine();

```

Update Operation

The following code shows an Update operation that targets the Employee table.

```
int result;

schemas.microsoft.com.Sql._2008._05.TableOp.dbo.Employee.RowPair updateRecordPair =
    new schemas.microsoft.com.Sql._2008._05.TableOp.dbo.Employee.RowPair();

schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee updateRecord =
    new schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee();

schemas.microsoft.com.Sql._2008._05.TableOp.dbo.Employee.RowPair[] updateArray =
    new schemas.microsoft.com.Sql._2008._05.TableOp.dbo.Employee.RowPair[1];

updateRecord = insertRecord[0];
updateRecord.Name = "Jeff Smith";

updateRecordPair.After = updateRecord;
updateRecordPair.Before = selectRecords[0];

updateArray[0] = updateRecordPair;

try
{
    Console.WriteLine("Updating the database...");
    result = client.Update(updateArray);
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("Updated Record for {0}", updateRecordPair.Before.Name);
Console.WriteLine("The new name for the employee is {0}", updateRecordPair.After.Name);
Console.WriteLine("Press any key to continue ...");
Console.ReadLine();
```

Delete Operation

The following code shows a Delete operation that targets the Employee table.

```
int deleteSuccess;

schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] deleteRecords =
    new schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[1];

deleteRecords = client.Select("*", "where [Employee_ID] = (select IDENT_CURRENT('Employee')
)");

Console.WriteLine("Following employees will be deleted from the database:");
for (int i = 0; i < deleteRecords.Length; i++)
{
    Console.WriteLine("Name: {0}", deleteRecords[i].Name);
}
Console.WriteLine("Press any key to begin deletion...");
Console.ReadLine();

try
{
    Console.WriteLine("Deleting employee record...");
    deleteSuccess = client.Delete(deleteRecords);
}

catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
}
```

```
    throw;  
}
```

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing Operations on Tables and Views with Large Data Types by Using the WCF Service Model

The SQL adapter enables adapter clients to read and update data in columns of large data types, that is, `varchar(max)`, `nvarchar(max)`, or `varbinary(max)`. To read data from such columns, adapter clients can use the `Select` operation. To insert or update data into such columns, the adapter exposes a `Set<column_name>` operation, where `<column_name>` is the name of the column of type `varchar(max)`, `nvarchar(max)`, or `varbinary(max)`.

Additionally, in SQL Server 2008, you can have the `varbinary(max)` column store unstructured data such as text documents and images. Such unstructured data is called FILESTREAM data. FILESTREAM data can be stored as files on the file system. The SQL adapter enables the client to enter FILESTREAM data into columns of type `varbinary(max)`. For more information about FILESTREAM storage, see <http://go.microsoft.com/fwlink/?LinkId=122485>.

This topic provides information about certain tasks you must perform on the computer running SQL Server and the computer running the adapter client to be able to insert or update FILESTREAM data. This topic also provides instructions on performing `Set<column_name>` operations to insert FILESTREAM data.

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

Prerequisites

You must perform the following tasks on the computer running SQL Server 2008 and the computer running the adapter client.

Note

If you are performing `Set<column_name>` operations on SQL Server 2005, you do not need to perform these tasks. You can directly start with the next section. This is because FILESTREAM is not supported on SQL Server 2005.

• On the computer running SQL Server 2008

- You must enable FILESTREAM on the SQL Server instance. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122486>.
- You must create a FILESTREAM-enabled database. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122487>.
- You must have a table for storing FILESTREAM data. For more information, see <http://go.microsoft.com/fwlink/?LinkId=122488>.

• On the computer running the adapter client

- You must have the SQL Client Connectivity SDK installed. You can install the SQL Client Connectivity SDK by running the SQL Server 2008 setup and selecting **SQL Client Connectivity SDK** in the **Feature Selection** page of the wizard. The adapter uses the `sqlncli10.dll`, installed with the SQL Client Connectivity SDK, to perform FILESTREAM operations.

After you have completed these tasks, you are all set to insert or update FILESTREAM data in SQL Server 2008 database tables.

How This Topic Demonstrates Operations on Large Data Types

To demonstrate how to perform `Set<column_name>` operations on tables with large data types, take a table, **Records**, that has columns **Id** and **Document**:

- The **Records** table, with all the data, is created by running the SQL script provided with the samples. For more information, see [Samples](#).

- The **Id** column is of type uniqueidentifier and takes a GUID. Assume that the **Id** column already has a GUID '438B7B4C-5491-409F-BCC1-78817C399EC3'.
- The **Document** column is of type VARBINARY(MAX). To update the **Document** column, the adapter exposes the **SetDocument** operation.

Note

For SQL Server 2008, to demonstrate FILESTREAM operations, assume that the **Document** column can store FILESTREAM data.

About the Examples Used in this Topic

The example in this topic performs operations on the **Records** table. The **Records** table is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **Records_FILESTREAM_Op**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Client Class

The name of the WCF client generated for the operations on large data types that the SQL adapter discovers, is based on the name of the table or view, as listed in the following table.

SQL Server Database Artifact	WCF Client Name
Table	TableOp_[Schema]_[TABLE_NAME]Client
View	ViewOp_[Schema]_[VIEW_NAME]Client

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

Method Signature for Invoking Operations on Columns of Large Data Types

The following table shows the method signatures for the basic operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Set<column_name>	public void Set<column_name>(string Filter, byte[] Data);

<column_name> = Name of the column of large data type.

As an example, the following code shows the method signatures for a WCF client class generated for the **SetDocument** operation on the **Records** table under the default "dbo" schema.

```
public partial class TableOp_dbo_RecordsClient : System.ServiceModel.ClientBase<TableOp_dbo_Records>, TableOp_dbo_Records {
    public void SetDocument (string Filter, byte[] Data);
}
```

In this snippet, **TableOp_dbo_RecordsClient** is the name of the WCF class in the SqlAdapterBindingClient.cs generated by the Add Adapter Service Reference Plug-in.

Parameters for Operations on Columns of Large Data Types

This section provides the parameters required by the Set<column_name> operation.

Parameter name	Description
string Filter	Specifies the WHERE clause based on which the adapter updates the record for the column of large data type.
byte[] Data	Specifies the value that must be updated for the column of large data type.

The `Set<column_name>` operation does not return any values.

Creating a WCF Client to Invoke Operations on Columns of Large Data Types

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **SetDocument** operation on the **Records** table. The adapter exposes the **SetDocument** operation to update data in columns of large data types.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **SetDocument** operation on the **Records** table. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).
3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql`, `Microsoft.ServiceModel.Channels`, and `System.Transactions`.
4. Open the `Program.cs` file and add the `System.Transactions` namespace.
5. In the `Program.cs`, create a client as described in the snippet below.

```
TableOp_dbo_RecordsClient client = new TableOp_dbo_RecordsClient("SqlAdapterBinding_TableOp_dbo_Records");
client.ClientCredentials.UserName.UserName = "";
client.ClientCredentials.UserName.Password = "";
```

In this snippet, `TableOp_dbo_RecordsClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_TableOp_dbo_Records` is the name of the client endpoint configuration and is defined in the `app.config`. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

Caution

To perform operations on FILESTREAM data, you must always connect to SQL Server using Windows authentication. To connect using Windows authentication, you must provide empty username and password, as shown in the preceding snippet. Also, before using Windows authentication to connect to SQL Server, you must have performed the steps mentioned in [Connecting to SQL Server Using Windows Authentication](#).

Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the SQL Adapter](#).

6. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

7. Invoke the **SetDocument** operation on the **Records** table.

⚠Caution

The Set<*column_name*> operations must always be performed in a transaction. To ensure this, the Set<*column_name*> operation must be invoked within a transaction scope and the **UseAmbientTransaction** binding property must be set to **true** in the app.config.

```
using (TransactionScope tx = new TransactionScope())
{
    string filter = "WHERE Id='438B7B4C-5491-409F-BCC1-78817C399EC3'";
    byte[] data = ASCIIEncoding.ASCII.GetBytes("Sample data");
    client.SetDocument(filter, data);
    tx.Complete();
}
```

Here, the application converts the string "Sample data" into a base64 encoded string, and updates it in the record that satisfies the filter criteria.

8. Close the client as described in the snippet below:

```
client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();
```

9. Build the project and then run it. The application updates the **Document** column in the **Records** table.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Invoking Stored Procedures Using the WCF Service Model

The SQL adapter discovers the stored procedures as operations that the adapter clients can invoke on the WCF client to invoke the stored procedure. Based on how the stored procedure returns the result set, the adapter categorizes all the stored procedures as:

- **Procedures.** Invoking stored procedures from the **Procedures** node returns an array of DataSet.
- **Strongly-Typed Procedures.** Invoking stored procedures from the **Strongly-Typed Procedures** node returns a strongly-typed result set.

Note that the same set of stored procedures in the database you connected to will be listed both under the **Procedures** and **Strongly-Typed Procedures** node. Based on the kind of result set you want, you must invoke the stored procedure from the relevant node. For more information about how the SQL adapter supports stored procedures, see [Executing Stored Procedures in SQL Server](#).

This section provides instructions on how to invoke stored procedures from both the **Procedures** and **Strongly-Typed Procedures** node by using a WCF client.

In This Section

- [Invoking Weakly-typed Stored Procedures Using the WCF Service Model](#)
- [Invoking Strongly-typed Stored Procedure Using WCF Service Model](#)

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Invoking Weakly-typed Stored Procedures Using the WCF Service Model

When you invoke a procedure listed under the **Procedures** node in the Add Adapter Service Reference Plug-in, the output is in the form of a DataSet array. This topic provides instructions on how to create a WCF client to invoke a stored procedure in SQL Server that returns a DataSet array.

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

About the Examples Used in this Topic

The example in this topic uses the GET_EMP_DETAILS stored procedure. This stored procedure takes an employee ID as an input parameter and returns all the corresponding columns for the employee with that ID. The GET_EMP_DETAILS stored procedure is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **Execute StoredProc**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Client Class

The name of the WCF client generated for invoking stored procedures under the **Procedures** node using the SQL adapter is listed in the following table.

SQL Server Database Artifact	WCF Client Name
Procedure (under the Procedures node)	Procedures_[schema]Client

[schema] is the schema to which the procedure belongs; for example "dbo".

Method Signature for Invoking Stored Procedures

The following table shows the signature for the method exposed to invoke the stored procedures.

Operation	Method Signature
Procedure name	System.Data.DataSet[] [procedure_name](param1, param2, ...)

[procedure_name] is the name of the procedure; for example GET_EMP_DETAILS

As an example, the signature for the method to invoke the GET_EMP_DETAILS procedure is shown in the following code snippet.

```
public partial class Procedures_dboClient : System.ServiceModel.ClientBase<Procedures_dbo>,
Procedures_dbo {
    public System.Data.DataSet[] GET_EMP_DETAILS(System.Nullable<int> emp_id, out int ReturnV
alue);
}
```

In this snippet,

- `Procedures_dboClient` is the name of the WCF client class. In this example, you use this class to create a client to invoke the stored procedure.
- `public System.Data.DataSet[] GET_EMP_DETAILS(System.Nullable<int> emp_id, out int ReturnValue)` is the method that you invoke in this example to invoke the stored procedure. This stored procedure takes an employee ID and returns a DataSet array.

Creating a WCF Client to Invoke a Stored Procedure in SQL Server

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section specifically describes how to create a WCF client

that invokes a stored procedure, the result set for which is a DataSet array. In this topic, as an example, you invoke the GET_EMP_DETAILS stored procedure. This stored procedure is created by running the SQL script provided with each sample.

To create a WCF client to invoke a stored procedure

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the GET_EMP_DETAILS stored procedure. Make sure you select the procedure under the **Procedures** node. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and create a client as described in the snippet below.

```
Procedures_dboClient client = new Procedures_dboClient("SqlAdapterBinding_Procedures_dbo");

client.ClientCredentials.UserName.UserName = "<Enter username here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `Procedures_dboClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_Procedures_dbo` is the name of the client endpoint configuration and is defined in the app.config. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the GET_EMP_DETAILS stored procedure. Before you invoke the GET_EMP_DETAILS procedure, you must add the `System.Data` namespace to your code.

```
DataSet[] dataArray;
int returnCode;

try
{
    Console.WriteLine("Calling a stored procedure...");
```

```

        dataArray = client.GET_EMP_DETAILS(10001, out returnCode); //Invoke the stored pro
cedure
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception: " + ex.Message);
        throw;
    }
    Console.WriteLine("The details for the employee with ID '10001' are:");
    Console.WriteLine("*****");

    foreach (DataSet dataSet in dataArray)
    {
        foreach (DataTable tab in dataArray[0].Tables)
        {
            foreach (DataRow row in tab.Rows)
            {
                for (int i = 0; i < tab.Columns.Count; i++)
                {
                    Console.WriteLine(row[i]);
                }
            }
        }
    }
    Console.WriteLine("*****");

```

7. Close the client as described in the snippet below:

```

client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();

```

8. Build the project and then run it. The details for the employee, for which you provided the ID, are displayed on the console.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Invoking Strongly-typed Stored Procedure Using WCF Service Model

When you invoke a procedure listed under the **Strongly-Typed Procedures** node in the Add Adapter Service Reference Plug-in, the output is in the form of a strongly-typed result set. This topic provides instructions on how to create a WCF client to invoke stored procedures in SQL Server that return a strongly-typed result set.

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) before you start developing your application.

About the Examples Used in this Topic

The example in this topic uses the GET_EMP_DETAILS stored procedure. This stored procedure takes an employee ID as an input parameter and returns all the corresponding columns for the employee with that ID. The GET_EMP_DETAILS stored procedure is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **Execute_TypedStoredProcedure**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Client Class

The name of the WCF client generated for invoking stored procedures under the **Strongly-Typed Procedures** node using the SQL adapter is listed in the following table.

SQL Server Database Artifact	WCF Client Name
Procedure (under the Strongly-Typed Procedures node)	TypedProcedures_[schema]Client

[schema] is the schema to which the procedure belongs; for example "dbo".

Method Signature for Invoking Stored Procedures

The following table shows the signature for the method exposed to invoke the stored procedures.

Operation	Method Signature
Procedure name	[PROC_NS] [procedure_name](param1, param2, ...)

[PROC_NS] is the procedure namespace; for example schemas.microsoft.com.Sql._2008._05.ProceduresResultSets.dbo.GET_EMP_DETAILS.StoredProcedureResultSet0[]

[procedure_name] is the name of the procedure; for example GET_EMP_DETAILS

As an example, the signature for the method to invoke the GET_EMP_DETAILS procedure is shown in the following code snippet.

```
public partial class TypedProcedures_dboClient : System.ServiceModel.ClientBase<TypedProcedures_dbo>, TypedProcedures_dbo{
    public schemas.microsoft.com.Sql._2008._05.ProceduresResultSets.dbo.GET_EMP_DETAILS.StoredProcedureResultSet0[]
        GET_EMP_DETAILS(System.Nullable<int> emp_id, out int ReturnValue);
}
```

In this snippet,

- TypedProcedures_dboClient is the name of the class. In this example, you use this class to create a client to invoke the stored procedure.
- public schemas.microsoft.com.Sql._2008._05.ProceduresResultSets.dbo.GET_EMP_DETAILS.StoredProcedureResultSet0[] GET_EMP_DETAILS(System.Nullable<int> emp_id, out int ReturnValue) is the method that you invoke in this example to invoke the stored procedure. This stored procedure takes an employee ID and returns a strongly-typed result

set.

Creating a WCF Client to Invoke a Stored Procedure in SQL Server

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section specifically describes how to create a WCF client that invokes a stored procedure, the result set for which is strongly-typed. In this topic, as an example, you invoke the GET_EMP_DETAILS stored procedure. This stored procedure is created by running the SQL script provided with each sample.

To create a WCF client to invoke a stored procedure

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the GET_EMP_DETAILS stored procedure. Make sure you select the procedure under the **Strongly-Typed Procedures** node. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

◆ Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and create a client as described in the snippet below.

```
TypedProcedures_dboClient client = new TypedProcedures_dboClient("SqlAdapterBinding_TypedProcedures_dbo");
client.ClientCredentials.UserName.UserName = "<Enter username here>";
client.ClientCredentials.UserName.Password = "<Enter username here>";
```

In this snippet, `TypedProcedures_dboClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_TypedProcedures_dbo` is the name of the client endpoint configuration and is defined in the `app.config`. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

📌 Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the GET_EMP_DETAILS stored procedure as described in the snippet below.

```
// Create array of type as specified in the method signature
schemas.microsoft.com.Sql._2008._05.ProceduresResultSets.dbo.GET_EMP_DETAILS.StoredPro
```

```

cedureResultSet0[] resultSet =
    new schemas.microsoft.com.Sql._2008._05.ProceduresResultSets.dbo.GET_EMP_DETAILS.St
oredProcedureResultSet0[1];
int returnCode;

try
{
    Console.WriteLine("Calling a stored procedure...");
    resultSet = client.GET_EMP_DETAILS(10001, out returnCode); //Invoke the stored pro
cedure
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
Console.WriteLine("The details for the employee with ID '10001' are:");
Console.WriteLine("*****");

for (int i = 0; i < resultSet.Length; i++)
{
    Console.WriteLine("Employee Name      : " + resultSet[i].Name);
    Console.WriteLine("Employee Designation : " + resultSet[i].Designation);
    Console.WriteLine("Employee Salary      : " + resultSet[i].Salary);
}

Console.WriteLine("*****");

```

7. Close the client as described in the snippet below:

```

client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();

```

8. Build the project and then run it. The name, designation, and salary for the employee ID, are displayed on the console.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Invoking Scalar Functions in SQL Server by Using the WCF Service Model

You can use the SQL adapter in a .NET application using the WCF service model to invoke scalar functions in SQL Server. The adapter exposes the scalar functions as methods that can be invoked directly on SQL Server. For more information about how the adapter supports scalar functions, see [Executing Scalar Functions in SQL Server](#).

How This Topic Demonstrates Invoking Scalar Functions Using the WCF Service Model

This topic demonstrates how to invoke the GET_EMP_ID function in a SQL Server database. The GET_EMP_ID function takes the designation of an employee in the **Employee** table and returns the corresponding employee ID. The GET_EMP_ID function and the **Employee** table are created by running the SQL script provided with the samples. For more information, see [Samples](#).

About the Examples Used in this Topic

The example in this topic invoked the GET_EMP_ID scalar function on the Employee table. The GET_EMP_ID function and the **Employee** table are created by running the SQL script provided with the samples. A sample, **ScalarFunction_ServiceModel**, which is based on this topic, is also provided with the SQL adapter samples. For more information, see [Samples](#).

The WCF Client Class

The name of the WCF client generated for invoking the scalar function in SQL Server using the SQL adapter is listed in the following table.

SQL Server Database Artifact	WCF Client Name
Scalar function	ScalarFunctions_[SCHEMA]Client

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

Method Signature for Invoking Scalar Functions

The following table shows the method signatures for the basic operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Scalar function name	public <return_type> <scalar_function_name>(param1, param2, ...)

<return_type> = Return type defined in the function definition

<scalar_function_name> = Name of the scalar function.

As an example, the following code shows the method signatures for a WCF client class generated for the **GET_EMP_ID** scalar functions, in the dbo schema, which takes the employee designation as a parameter and returns an employee ID (integer).

```
public partial class ScalarFunctions_dboClient : System.ServiceModel.ClientBase<ScalarFunctions_dbo>, ScalarFunctions_dbo {
    public System.Nullable<int> GET_EMP_ID(string emp_desig);
}
```

In this snippet, **ScalarFunctions_dboClient** is the name of the WCF class in the SqlAdapterBindingClient.cs generated by the Add Adapter Service Reference Plug-in.

Parameters for Invoking Scalar Functions

The parameters for the methods exposed by the SQL adapter to invoke a scalar function are the same as the parameters defined in the scalar function definition in SQL Server. For example, the parameter for invoking the GET_EMP_ID scalar function is emp_desig and takes an employee's designation.

Again, the return value for a scalar function is same as the return value defined in the scalar function definition in SQL Server. For example, the return value for the GET_EMP_ID function is an employee's ID of type integer.

Creating a WCF Client to Invoke Scalar Functions

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described

in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **GET_EMP_ID** scalar function.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **GET_EMP_ID** scalar function. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).
3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs and create a client as described in the snippet below.

```
ScalarFunctions_dboClient client = new ScalarFunctions_dboClient("SqlAdapterBinding_Sc
alarFunctions_dbo");
client.ClientCredentials.UserName.UserName = "<Enter user name here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `ScalarFunctions_dboClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_ScalarFunctions_dbo` is the name of the client endpoint configuration and is defined in the app.config. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying the client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the **GET_EMP_ID** function to retrieve the ID for an employee with the designation as "Manager".

```
Console.WriteLine("Invoking the GET_EMP_ID function");
string emp_designation = "Manager";
try
{
    System.Nullable<int> emp_id = client.GET_EMP_ID(emp_designation);
    Console.WriteLine("The Employee ID for the employee with 'Manager' designation is: " + emp_id);
}
catch (Exception e)
{
    Console.WriteLine("Exception: " + e.Message);
    throw;
}
```

Note

For the sake of simplicity, the **Employee** table has only one employee with “Manager” designation. If your target table has more employees with the same designation, you must define the function accordingly.

7. Close the client as described in the snippet below:

```
client.Close();  
Console.WriteLine("Press any key to exit...");  
Console.ReadLine();
```

8. Build the project and then run it. The application displays the employee ID of the employee with the designation of “Manager”.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Invoking Table-Valued Functions in SQL Server by Using the WCF Service Model

You can use the SQL adapter in a .NET application using the WCF service model to invoke table-valued functions in SQL Server. The adapter exposes the table-valued functions as methods that can be invoked directly on SQL Server. For more information about how the adapter supports scalar functions, see [Executing Table-Valued Functions in SQL Server](#).

How This Topic Demonstrates Invoking Scalar Functions Using the WCF Service Model

This topic demonstrates how to invoke the TVF_EMPLOYEE function in a SQL Server database. The TVF_EMPLOYEE function takes the designation of an employee in the **Employee** table and returns the record for the employee. The TVF_EMPLOYEE function and the **Employee** table are created by running the SQL script provided with the samples. For more information, see [Samples](#).

About the Examples Used in this Topic

The example in this topic invoked the TVF_EMPLOYEE table-valued function on the **Employee** table. TVF_EMPLOYEE function and the **Employee** table are created by running the SQL script provided with the samples. A sample, **TableFunction_ServiceModel**, which is based on this topic, is also provided with the SQL adapter samples. For more information, see [Samples](#).

The WCF Client Class

The name of the WCF client generated for invoking the scalar function in SQL Server using the SQL adapter is listed in the following table.

SQL Server Database Artifact	WCF Client Name
Table-valued function	TableValuedFunctions_[SCHEMA]Client

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

Method Signature for Invoking Table-valued Functions

The following table shows the method signatures for the basic operations on a table. The signatures are the same for a view, except that the view namespace and name replace those of the table.

Operation	Method Signature
Table-valued function name	public [NAMESPACE][FUNCTION_NAME][] [FUNCTION_NAME](param1, param2, ...)

[NAMESPACE] = The namespace, for example, schemas.microsoft.com.Sql_2008._05.Types.TableFunctionReturnTables.dbo.TVF_EMPLOYEE

[FUNCTION_NAME] = Name of the table-valued function.

As an example, the following code shows the method signatures for a WCF client class generated for the **TVF_EMPLOYEE** scalar functions, in the dbo schema, which takes the employee designation as a parameter and returns the employee record.

```
public partial class TableValuedFunctions_dboClient : System.ServiceModel.ClientBase<TableValuedFunctions_dbo>, TableValuedFunctions_dbo {
    public schemas.microsoft.com.Sql_2008._05.Types.TableFunctionReturnTables.dbo.TVF_EMPLOYEE[] TVF_EMPLOYEE(string emp_desig);
}
```

In this snippet, **TableValuedFunctions_dboClient** is the name of the WCF class in the SqlAdapterBindingClient.cs generated by the Add Adapter Service Reference Plug-in.

Parameters for Invoking Table-valued Functions

The parameters for the methods exposed by the SQL adapter to invoke a table-valued function are the same as the parameters defined in the function definition in SQL Server. For example, the parameter for invoking the TVF_EMPLOYEE table-valued function is emp_desig and takes an employee's designation.

Again, the return value for a table-valued function is same as the return value defined in the function definition in SQL Server.

For example, the return value for the TVF_EMPLOYEE function is an array of records of type `schemas.microsoft.com.Sql._2008._05.Types.TableFunctionReturnTables.dbo.TVF_EMPLOYEE[]`.

Creating a WCF Client to Invoke Table-valued Functions

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section describes how to create a WCF client to invoke the **TVF_EMPLOYEE** table-valued function.

To create a WCF client

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **TVF_EMPLOYEE** scalar function. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).
3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs and create a client as described in the snippet below.

```
TableValuedFunctions_dboClient client = new TableValuedFunctions_dboClient("SqlAdapterBinding_TableValuedFunctions_dbo");
client.ClientCredentials.UserName.UserName = "<Enter user name here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `TableValuedFunctions_dboClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_TableValuedFunctions_dbo` is the name of the client endpoint configuration and is defined in the app.config. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying then client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the **TVF_EMPLOYEE** function to retrieve all the employee records having the "Manager" designation.

```
Console.WriteLine("Invoking the TVF_EMPLOYEE function");
schemas.microsoft.com.Sql._2008._05.Types.TableFunctionReturnTables.dbo.TVF_EMPLOYEE[]
emp_details;
string emp_designation = "Manager";

try
{
    emp_details = client.TVF_EMPLOYEE(emp_designation);
}
catch (Exception e)
```

```
{
    Console.WriteLine("Exception: " + e.Message);
    throw;
}
Console.WriteLine("The details for the employee with the 'Manager' designation are:");
Console.WriteLine("*****");
for (int i = 0; i < emp_details.Length; i++)
{
    Console.WriteLine("Employee ID      : " + emp_details[i].Employee_ID);
    Console.WriteLine("Employee Name    : " + emp_details[i].Name);
    Console.WriteLine("Employee Designation: " + emp_details[i].Designation);
    Console.WriteLine("Employee Salary   : " + emp_details[i].Salary);
    Console.WriteLine();
}
}
```

7. Close the client as described in the snippet below:

```
client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();
```

8. Build the project and then run it. The application displays the employee ID, name, and salary of all the employees with a "Manager" designation.

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using WCF Service Model

The SQL adapter exposes generic SQL Server operations such as **ExecuteNonQuery**, **ExecuteReader**, and **ExecuteScalar**. You can use these operations to execute any SQL statement on a SQL Server database. These operations differ based on the kind of response you get for the SQL statement. For more information about how the adapter supports these operations, see [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).

This topic demonstrates how to perform an **ExecuteReader** operation using the SQL adapter using the WCF service model. You can follow the same set of procedures described in this topic to perform **ExecuteNonQuery** and **ExecuteScalar** operations.

About the Examples Used in this Topic

The example in this topic uses the **ExecuteReader** operation to execute the ADD_EMP_DETAILS stored procedure. This stored procedure adds a record to the Employee table and returns the employee ID for the record. The ADD_EMP_DETAILS stored procedure is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **ExecuteReader**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Client Class

The name of the WCF client generated for invoking generic operations (ExecuteNonQuery, ExecuteReader, or ExecuteScalar) using the SQL adapter is listed in the following table.

Operations	WCF Client Name
ExecuteNonQuery, ExecuteReader, or ExecuteScalar	GenericTableOpClient

Method Signature for Invoking Generic Operations

The following table shows the signature for the method exposed to invoke the generic operations.

Operation	Method Signature
ExecuteNonQuery	int ExecuteNonQuery(string Query)
ExecuteReader	System.Data.DataSet[] ExecuteReader(string Query)
ExecuteScalar	string ExecuteScalar(string Query)

As an example, the signature for the generic operation methods is shown in the following code snippet.

```
public partial class GenericTableOpClient : System.ServiceModel.ClientBase<GenericTableOp>,
GenericTableOp {
    public int ExecuteNonQuery(string Query);
    public System.Data.DataSet[] ExecuteReader(string Query);
    public string ExecuteScalar(string Query);
}
```

In this snippet,

- `GenericTableOpClient` is the name of the class. In this example, you use this class to create a client to invoke the generic operation, `ExecuteReader`.
- `public System.Data.DataSet[] ExecuteReader(string Query)` is the method that you invoke in this example to invoke the ADD_EMP_DETAILS stored procedure.

Creating a WCF Client to Invoke an ExecuteReader Operation

The generic set of actions required to perform an operation on SQL Server using a WCF client involves a set of tasks described in [Overview of Using the WCF Service Model with the Adapter](#). This section specifically describes how to create a WCF client

that invokes an **ExecuteReader** operation to execute the ADD_EMP_DETAILS stored procedure. This stored procedure is created by running the SQL script provided with each sample.

To create a WCF client to invoke ExecuteReader operation

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. Generate the WCF client class for the **ExecuteReader** generic operation. This operation is available under the root node when you connect to the SQL Server database using the Add Adapter Service Reference Plug-in. For more information about generating a WCF client class, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

◆ Important

Before generating the WCF client class, make sure you set the **EnableBizTalkCompatibilityMode** binding property to false.

3. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql` and `Microsoft.ServiceModel.Channels`.
4. Open the Program.cs file and create a client as described in the snippet below.

```
GenericTableOpClient client = new GenericTableOpClient("SqlAdapterBinding_GenericTableOp");
client.ClientCredentials.UserName.UserName = "<Enter username here>";
client.ClientCredentials.UserName.Password = "<Enter password here>";
```

In this snippet, `GenericTableOpClient` is the WCF client defined in `SqlAdapterBindingClient.cs`. This file is generated by the Add Adapter Service Reference Plug-in. `SqlAdapterBinding_GenericTableOp` is the name of the client endpoint configuration and is defined in the `app.config`. This file is also generated by the Add Adapter Service Reference Plug-in and contains the binding properties and other configuration settings.

📌 Note

In this snippet, you use the binding and endpoint address from the configuration file. You can also explicitly specify these values in your code. For more information on the different ways of specifying client binding, see [Specifying a Client Binding for the SQL Adapter](#).

5. Open the client as described in the snippet below:

```
try
{
    Console.WriteLine("Opening Client...");
    client.Open();
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
```

6. Invoke the **ExecuteReader** operation for the ADD_EMP_DETAILS stored procedure. Before you invoke the `ExecuteReader` operation, you must add the `System.Data` namespace to your code.

```
string query = "EXEC ADD_EMP_DETAILS 'Tom Smith', 'Manager', 500000";
DataSet[] dsArray = client.ExecuteReader(query);

Console.WriteLine("Invoking the ADD_EMP_DETAILS stored procedure using ExecuteReader");
;
Console.WriteLine("*****");
foreach (DataSet dataSet in dsArray)
```

```
{
    foreach (DataTable tab in dsArray[0].Tables)
    {
        foreach (DataRow row in tab.Rows)
        {
            for (int i = 0; i < tab.Columns.Count; i++)
            {
                Console.WriteLine("The ID for the newly added employee is : " + row[i]);
            }
        }
    }
}
Console.WriteLine("*****");
```

7. Close the client as described in the snippet below:

```
client.Close();
Console.WriteLine("Press any key to exit...");
Console.ReadLine();
```

8. Build the project and then run it. The employee ID of the newly inserted employee is displayed on the console.

Polling SQL Server by Using the SQL Adapter with WCF Service Model

You can configure the SQL adapter to receive polling-based data-changed messages from SQL Server. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set. Based on the type of polling message received, the adapter exposes different polling operations:

- **Polling.** This operation returns a data set as part of the polling message.
- **TypedPolling.** This operation returns a strongly-typed polling message.
- **XmlPolling.** This operation returns the polling message as an XML message. You must use this operation if you want to use SELECT statements or stored procedures that use the FOR XML clause to return data as XML messages. For more information about the FOR XML clause, see <http://go.microsoft.com/fwlink/?LinkId=131402>.

For more information about these polling operations, see [Support for Polling](#).

Note
The SQL adapter enables adapter clients to have a single application with more than one Polling or TypedPolling operations for the same database or table. To support such a scenario, the adapter includes a unique ID— InboundID —in the connection URI. This ID, when added to the connection URI, makes it unique, thereby enabling multiple polling operations in a single application.

The topics in this section provide instructions on how to use both Polling and TypedPolling operations in a .NET application.

In This Section

- [Receiving Polling-based Data-changed Messages from SQL Server by Using the WCF Service Model](#)
- [Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using WCF Service Model](#)

See Also

Other Resources

[Developing Applications by Using the WCF Service Model](#)

Receiving Polling-based Data-changed Messages from SQL Server by Using the WCF Service Model

You can configure the SQL adapter to receive periodic data-change messages for SQL Server tables or views. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set.

For more information on how the adapter supports polling, see [Support for Polling](#).

Note

This topic demonstrates how to use the **Polling** inbound operation to use polling messages. The message for the Polling operation is not strongly-typed. If you want to get strongly-typed polling message, you must use the **TypedPolling** operation. You must also use the **TypedPolling** operation to have multiple polling operations in a single application. For instructions on how to perform **TypedPolling** operation, see

[Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using WCF Service Model](#).

Important

If you want to have more than one polling operation in a single application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. The inbound ID you specify is added to the operation namespace to make it unique.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the SQL adapter supports receiving data change messages, create a .NET application and generate the WCF service contract for the **Polling** operation. If you want to specify the polling related binding properties while generating the WCF service contract, specify the **PolledDataAvailableStatement** as:

```
SELECT COUNT(*) FROM Employee
```

The **PolledDataAvailableStatement** must return a result set with the first cell containing a positive value. If the first cell does not contain a positive value, the adapter does not execute the polling statement.

As part of the polling statement, perform the following operations:

1. Select all the rows from the Employee table.
2. Execute a stored procedure (MOVE_EMP_DATA) to move all the records from the Employee table to an EmployeeHistory table.
3. Execute a stored procedure (ADD_EMP_DETAILS) to add a new record to the Employee table. This procedure takes the employee name, designation, and salary as parameters.

To perform these operations, you must specify the following for the **PollingStatement** binding property:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000
```

After the polling statement is executed, all the records from the Employee table are selected and the message from SQL Server is received. After the MOVE_EMP_DATA stored procedure is executed by the adapter, all the records are moved to the EmployeeHistory table. Then, the ADD_EMP_DETAILS stored procedure is executed to add a new record to the Employee table. The next polling execution will only return a single record. This cycle continues until you close the service host.

Configuring a Polling Query with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure the adapter to receive data-change messages. You must specify these binding properties as part of the .NET application for polling.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling , TypedPolling , or Notification inbound operation. Default is Polling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. The SQL statement must return a result set consisting of rows and columns. Only if a row is available, the SQL statement specified for the PollingStatement binding property will be executed.
PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter waits for the remaining time in the interval.
PollingStatement	Specifies the SQL statement to poll the SQL Server database table. You can specify a simple SELECT statement or a stored procedure for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property. You can specify any number of SQL statements separated by a semi-colon.
PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false .

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to poll SQL Server, read further.

Configuring Polling in the WCF Service Model

To receive the **Polling** operation when you use the WCF service model, you must:

1. Generate a WCF service contract (interface) for the **Polling** operation from the metadata exposed by the adapter. To do this, you could use the Add Adapter Service Reference Visual Studio Plug-in.
2. Implement a WCF service from this interface.
3. Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic poll the Employee table. The example also uses the MOVE_EMP_DATA and ADD_EMP_DETAILS stored procedure. A script to generate these artifacts is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **Polling_ServiceModel**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **Polling** operation. For more information about generating a WCF service contract, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **Polling** operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/", ConfigurationName="PollingOperation")]
```

```

public interface PollingOperation {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/Sql/2008/05/Polling/) of message Polling
    // does not match the default value (http://schemas.microsoft.com/Sql/2008/05/)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="Polling")]
    [System.ServiceModel.XmlSerializerFormatAttribute()]
    void Polling(Polling request);
}

```

The Message Contracts

The message contract namespace is modified by the **InboundID** parameter in the connection URI, if specified. In this example, you did not specify an inbound ID in the connection URI. The request message returns a DataSet.

```

[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="Polling", WrapperNamespace="http://schemas.microsoft.com/Sql/2008/05/Polling/", IsWrapped=true)]
public partial class Polling {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/Polling/", Order=0)]
    [System.Xml.Serialization.XmlArrayAttribute(IsNullable=true)]
    [System.Xml.Serialization.XmlArrayItemAttribute("DataSet", Namespace="http://schemas.datacontract.org/2004/07/System.Data", IsNullable=false)]
    public System.Data.DataSet[] PolledData;

    public Polling() {
    }

    public Polling(System.Data.DataSet[] PolledData) {
        this.PolledData = PolledData;
    }
}

```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is SqlAdapterBindingService.cs. You can insert the logic to process the **Polling** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```

namespace SqlAdapterBindingNamespace {

    public class SqlAdapterBindingService : PollingOperation {

        // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/Sql/2008/05/Polling/) of message Polling
        // does not match the default value (http://schemas.microsoft.com/Sql/2008/05/)
        public virtual void Polling(Polling request) {
            throw new System.NotImplementedException("The method or operation is not implemented.");
        }
    }
}

```

Receiving Inbound Messages for Polling Operation

This section provides instructions on how to write a .NET application to receive inbound polling messages using the SQL adapter.

To receive polling messages from the SQL adapter

1. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **Polling** operation. For more information, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

2. Implement a WCF service from the interface and helper classes generated in step 1. The **Polling** method of this class can throw an exception to abort the polling transaction, if an error is encountered processing the data received from the **Polling** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **Polling** method, you can implement your application logic directly. This class can be found in `SqlAdapterBindingService.cs`. This code in this example sub-classes the **SqlAdapterBindingService** class. In this code, the polling message received as a `DataSet` is written to the console.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class PollingService : SqlAdapterBindingNamespace.SqlAdapterBindingService
{

public override void Polling(Polling request)
{
    Console.WriteLine("\nNew Polling Records Received");
    Console.WriteLine("*****");
    DataSet[] dataArray = request.PolledData;
    foreach (DataTable tab in dataArray[0].Tables)
    {
        foreach (DataRow row in tab.Rows)
        {
            for (int i = 0; i < tab.Columns.Count; i++)
            {
                Console.WriteLine(row[i]);
            }
        }
    }
    Console.WriteLine("*****");
    Console.WriteLine("\nHit <RETURN> to stop polling");
}
}
```

3. Because the SQL adapter does not accept credentials as part of the connection URI, you must implement the following class to pass credentials for the SQL Server database. In the latter part of the application, you will instantiate this class to pass on the SQL Server credentials.

```
class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }
}
```

```

protected override ClientCredentials CloneCore()
{
    ClientCredentials clone = new PollingCredentials();
    clone.UserName.UserName = this.UserName.UserName;
    clone.UserName.Password = this.UserName.Password;
    return clone;
}
}

```

4. Create a **SqlAdapterBinding** and configure the polling operation by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType**, **PolledDataAvailableStatement**, and **PollingStatement**.

```

SqlAdapterBinding binding = new SqlAdapterBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";
binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DET
AILS John, Tester, 100000";

```

5. Specify SQL Server database credentials by instantiating the **PollingCredentials** class you created in Step 3.

```

PollingCredentials credentials = new PollingCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

```

6. Create an instance of the WCF service created in step 2.

```

// create service instance
PollingService service = new PollingService();

```

7. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. The base connection URI cannot contain the inbound ID, if specified. You should also specify the credentials here.

```

// Enable service host
Uri[] baseUri = new Uri[] { new Uri("mssql://mysqlserver//mydatabase") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);

```

8. Add a service endpoint to the service host. To do this:

- Use the binding created in step 4.
- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "PollingOperation".

```

// Add service endpoint: be sure to specify PollingOperation as the contract
Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");
serviceHost.AddServiceEndpoint("PollingOperation", binding, ConnectionUri);

```

9. To receive polling data, open the service host. The adapter will return data whenever the query returns a result set.

```

// Open the service host to begin polling

```

```
serviceHost.Open();
```

10. To terminate polling, close the service host.

◆ Important

The adapter will continue to poll until the service host is closed.

```
serviceHost.Close();
```

Example

The following example shows a polling query that executes the Employee table. The polling statement performs the following tasks:

1. Selects all the records from the Employee table.
2. Executes the MOVE_EMP_DATA stored procedure to move all records from Employee table to EmployeeHistory table.
3. Executes the ADD_EMP_DETAILS stored procedure to add a single record to the Employee table.

The first polling message will contain all the records from the Employee table. The subsequent polling messages will contain only the last record inserted by the ADD_EMP_DETAILS stored procedure. The adapter will continue to poll until you close the service host by pressing <RETURN>.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Microsoft.Adapters.Sql;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;
using System.Data;

namespace Polling_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class PollingService : SqlAdapterBindingNamespace.SqlAdapterBindingService
    {

        public override void Polling(Polling request)
        {
            Console.WriteLine("\nNew Polling Records Received");
            Console.WriteLine("*****");
            DataSet[] dataArray = request.PolledData;
            foreach (DataTable tab in dataArray[0].Tables)
            {
                foreach (DataRow row in tab.Rows)
                {
                    for (int i = 0; i < tab.Columns.Count; i++)
                    {
                        Console.WriteLine(row[i]);
                    }
                }
            }
            Console.WriteLine("*****");
            Console.WriteLine("\nHit <RETURN> to stop polling");
        }
    }
}
```

```

}

class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
dingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
tBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
HostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new PollingCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

class Program
{
    static void Main(string[] args)
    {
        ServiceHost serviceHost = null;
        try
        {
            Console.WriteLine("Sample started...");
            Console.WriteLine("Press any key to start polling...");
            Console.ReadLine();

            SqlAdapterBinding binding = new SqlAdapterBinding();
            binding.InboundOperationType = InboundOperation.Polling;
            binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";
            binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC
ADD_EMP_DETAILS John, Tester, 100000";
            Console.WriteLine("Binding properties assigned...");

            // This URI is used to specify the address for the ServiceEndpoint
            // It must contain the InboundId (if any) that was used to generate
            // the WCF service callback interface
            Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");

            // This URI is used to initialize the ServiceHost. It cannot contain
            // a query_string (InboundID); otherwise, an exception is thrown when
            // the ServiceHost is initialized.
            Uri[] baseUri = new Uri[] { new Uri("mssql://mysqlserver//mydatabase") };

            PollingCredentials credentials = new PollingCredentials();
            credentials.UserName.UserName = "<Enter user name here>";
            credentials.UserName.Password = "<Enter password here>";

            Console.WriteLine("Opening service host...");
            PollingService service = new PollingService();
            serviceHost = new ServiceHost(service, baseUri);
            serviceHost.Description.Behaviors.Add(credentials);
            serviceHost.AddServiceEndpoint("PollingOperation", binding, ConnectionUri);
            serviceHost.Open();
            Console.WriteLine("Service host opened...");
        }
        catch { }
    }
}

```


Receiving Strongly-typed Polling-based Data-changed Messages from SQL Server Using WCF Service Model

You can configure the SQL adapter to receive strongly-typed polling messages from SQL Server. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set. You must use strongly-typed polling in a scenario where you want to receive a strongly-typed result set. For more information on how the adapter supports strongly-typed polling, see [Support for Polling](#).

Important

If you want to have more than one polling operation in a single application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. The inbound ID you specify is added to the operation namespace to make it unique.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the SQL adapter supports receiving strongly-typed data change messages, create a .NET application and generate the WCF service contract for the **TypedPolling** operation. Make sure you specify the following while generating the WCF service contract:

- You must specify an **InboundID** as part of the connection URI.
- You must specify a polling statement for the **PollingStatement** binding property.

Additionally, if you want to specify other polling related binding properties while generating the proxy class, specify the **PolledDataAvailableStatement** as:

```
SELECT COUNT(*) FROM Employee
```

The **PolledDataAvailableStatement** must return a result set with the first cell containing a positive value. If the first cell does not contain a positive value, the adapter does not execute the polling statement.

As part of the polling statement, perform the following operations:

1. Select all the rows from the Employee table.
2. Execute a stored procedure (MOVE_EMP_DATA) to move all the records from the Employee table to an EmployeeHistory table.
3. Execute a stored procedure (ADD_EMP_DETAILS) to add a new record to the Employee table. This procedure takes the employee name, designation, and salary as parameters.

To perform these operations, you must specify the following for the **PollingStatement** binding property while generating the WCF service contract and helper classes:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000
```

After the polling statement is executed, all the records from the Employee table are selected and the message from SQL Server is received. After the MOVE_EMP_DATA stored procedure is executed by the adapter, all the records are moved to the EmployeeHistory table. Then, the ADD_EMP_DETAILS stored procedure is executed to add a new record to the Employee table. The next polling execution will only return a single record. This cycle continues until you close the service host.

Configuring Typed Polling with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure the adapter to receive data-change messages. Other than the **PollingStatement** binding property, all the other binding properties listed in this section are required while running the .NET application. You must specify the **PollingStatement** binding property before generating the WCF service contract **TypedPolling** operation.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling , TypedPolling , or Notification inbound operation. Default is Polling . To receive strongly-typed polling messages, set this to TypedPolling .
PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. The SQL statement must return a result set consisting of rows and columns. Only if a row is available, the SQL statement specified for the PollingStatement binding property will be executed.
PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter waits for the remaining time in the interval.
PollingStatement	Specifies the SQL statement to poll the SQL Server database table. You can specify a simple SELECT statement or a stored procedure for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property. You can specify any number of SQL statements separated by a semicolon. ♦ Important For TypedPolling , you must specify this binding property before generating metadata.
PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false .

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to poll SQL Server, read further.

Configuring Strongly-typed Polling in the WCF Service Model

To receive the **Polling** operation when you use the WCF service model, you must:

1. Generate a WCF service contract (interface) for the **TypedPolling** operation from the metadata exposed by the adapter. To do this, you could use the Add Adapter Service Reference Visual Studio Plug-in. While generating the WCF service contract for this example, make sure:

- You specify the **InboundID** as **Employee**.
- You specify a polling statement for the **PollingStatement** binding property. For this example, specify the polling statement as:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 10000
0
```

2. Implement a WCF service from this interface.
3. Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic poll the Employee table. The example also uses the MOVE_EMP_DATA and ADD_EMP_DETAILS stored procedure. A script to generate these artifacts is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **TypedPolling_ServiceModel**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for the **TypedPolling** operation. For more information about generating a WCF service contract, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **TypedPolling** operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/", ConfigurationName="TypedPolling_Employee")]
public interface TypedPolling_Employee {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/Sql/2008/05/TypedPolling/Employee) of message TypedPolling
    // does not match the default value (http://schemas.microsoft.com/Sql/2008/05/)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="TypedPolling")]
    void TypedPolling(TypedPolling request);
}
```

The Message Contracts

The message contract namespace is modified by the **InboundID** parameter in the connection URI, if specified. In this example, you specified the inbound ID as **Employee**. The request message returns a strongly-typed result set.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="TypedPolling", WrapperNamespace="http://schemas.microsoft.com/Sql/2008/05/TypedPolling/Employee", IsWrapped=true)]
public partial class TypedPolling {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/TypedPolling/Employee", Order=0)]
    public schemas.microsoft.com.Sql._2008._05.TypedPolling.Employee.TypedPollingResultSet0
    [] TypedPollingResultSet0;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/TypedPolling/Employee", Order=1)]
    public schemas.microsoft.com.Sql._2008._05.TypedPolling.Employee.TypedPollingResultSet1
    [] TypedPollingResultSet1;

    public TypedPolling() {
    }

    public TypedPolling(schemas.microsoft.com.Sql._2008._05.TypedPolling.Employee.TypedPollingResultSet0[] TypedPollingResultSet0, schemas.microsoft.com.Sql._2008._05.TypedPolling.Employee.TypedPollingResultSet1[] TypedPollingResultSet1) {
        this.TypedPollingResultSet0 = TypedPollingResultSet0;
        this.TypedPollingResultSet1 = TypedPollingResultSet1;
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is SqlAdapterBindingService.cs. You can insert the logic to process the **TypedPolling** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```
namespace SqlAdapterBindingNamespace {
```

```

public class SqlAdapterBindingService : TypedPolling_Employee {
    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas
.microsoft.com/Sql/2008/05/TypedPolling/Employee) of message TypedPolling
    // does not match the default value (http://schemas.microsoft.com/Sql/2008/05/)
    public virtual void TypedPolling(TypedPolling request) {
        throw new System.NotImplementedException("The method or operation is not implem
ented.");
    }
}

```

Receiving Strongly-typed Inbound Messages for Polling Operation

This section provides instructions on how to write a .NET application to receive strongly-typed inbound polling messages using the SQL adapter.

To receive polling messages from the SQL adapter

1. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **TypedPolling** operation. Make sure you specify the following while generating the WCF service contract for this example:
 - You must specify the **InboundID** as **Employee**.
 - You must specify a polling statement for the **PollingStatement** binding property. For this example, specify the polling statement as:

```

SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 10000
0

```

For more information, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

2. Implement a WCF service from the interface and helper classes generated in step 1. The **TypedPolling** method of this class can throw an exception to abort the polling transaction, if an error is encountered processing the data received from the **TypedPolling** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

```

Within the **TypedPolling** method, you can implement your application logic directly. This class can be found in `SqlAdapterBindingService.cs`. This code in this example sub-classes the **SqlAdapterBindingService** class. In this code, the polling message received as a strongly-typed result set is written to the console.

```

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class PollingService : SqlAdapterBindingNamespace.SqlAdapterBindingService
{
    public override void TypedPolling(TypedPolling request)
    {
        Console.WriteLine("\nNew Polling Records Received");
        Console.WriteLine("*****");
        Console.WriteLine("Employee ID\tName\tDesignation\tSalary");

        for (int i = 0; i < request.TypedPollingResultSet0.Length; i++)
        {
            Console.WriteLine("{0}\t{1}\t{2}\t{3}",

```

```

        request.TypedPollingResultSet0[i].Employee_ID,
        request.TypedPollingResultSet0[i].Name,
        request.TypedPollingResultSet0[i].Designation,
        request.TypedPollingResultSet0[i].Salary);
    }
    Console.WriteLine("*****");
    Console.WriteLine("\nHit <RETURN> to stop polling");
}
}

```

3. Because the SQL adapter does not accept credentials as part of the connection URI, you must implement the following class to pass credentials for the SQL Server database. In the latter part of the application, you will instantiate this class to pass on the SQL Server credentials.

```

class PollingCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new PollingCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

```

4. Create a **SqlAdapterBinding** and configure the polling operation by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType**, **PolledDataAvailableStatement**, and **PollingStatement**.

```

SqlAdapterBinding binding = new SqlAdapterBinding();
binding.InboundOperationType = InboundOperation.TypedPolling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";
binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000";

```

5. Specify SQL Server database credentials by instantiating the **PollingCredentials** class you created in Step 3.

```

PollingCredentials credentials = new PollingCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

```

6. Create an instance of the WCF service created in step 2.

```
// create service instance
PollingService service = new PollingService();
```

7. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. The base connection URI cannot contain the inbound ID. You must also specify the credentials here.

```
// Enable service host
Uri[] baseUri = new Uri[] { new Uri("mssql://mysqlserver//mydatabase") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);
```

8. Add a service endpoint to the service host. To do this:

- Use the binding created in step 4.
- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "TypedPolling_Employee".

```
// Add service endpoint: be sure to specify TypedPolling_Employee as the contract
Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?InboundID=Empliyee");
serviceHost.AddServiceEndpoint("TypedPolling_Employee", binding, ConnectionUri);
```

9. To receive polling data, open the service host. The adapter will return data whenever the query returns a result set.

```
// Open the service host to begin polling
serviceHost.Open();
```

10. To terminate polling, close the service host.

◆ Important

The adapter will continue to poll until the service host is closed.

```
serviceHost.Close();
```

Example

The following example shows a polling query that executes the Employee table. The polling statement performs the following tasks:

1. Selects all the records from the Employee table.
2. Executes the MOVE_EMP_DATA stored procedure to move all records from Employee table to EmployeeHistory table.
3. Executes the ADD_EMP_DETAILS stored procedure to add a single record to the Employee table.

The first polling message will contain all the records from the Employee table. The subsequent polling messages will contain only the last record inserted by the ADD_EMP_DETAILS stored procedure. The adapter will continue to poll until you close the service host by pressing <RETURN>.

```
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;

using Microsoft.Adapters.Sql;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace TypedPolling_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

    public class PollingService : SqlAdapterBindingNamespace.SqlAdapterBindingService
    {
        public override void TypedPolling(TypedPolling request)
        {
            Console.WriteLine("\nNew Polling Records Received");
            Console.WriteLine("*****");
            Console.WriteLine("Employee ID\tName\tDesignation\tSalary");

            for (int i = 0; i < request.TypedPollingResultSet0.Length; i++)
            {
                Console.WriteLine("{0}\t{1}\t{2}\t{3}",
                    request.TypedPollingResultSet0[i].Employee_ID,
                    request.TypedPollingResultSet0[i].Name,
                    request.TypedPollingResultSet0[i].Designation,
                    request.TypedPollingResultSet0[i].Salary);
            }
            Console.WriteLine("*****");
            Console.WriteLine("\nHit <RETURN> to stop polling");
        }

        class PollingCredentials : ClientCredentials, IServiceBehavior
        {
            public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
            Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
            dingParameters)
            {
                bindingParameters.Add(this);
            }

            public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
            tBase serviceHostBase)
            { }

            public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
            HostBase)
            { }

            protected override ClientCredentials CloneCore()
            {
                ClientCredentials clone = new PollingCredentials();
                clone.UserName.UserName = this.UserName.UserName;
                clone.UserName.Password = this.UserName.Password;
                return clone;
            }
        }

        class Program
        {
            static void Main(string[] args)
            {
                ServiceHost serviceHost = null;
                try
                {
                    Console.WriteLine("Sample started...");
                }
            }
        }
    }
}

```


Receiving Query Notifications Using the WCF Service Model

This topic demonstrates how to configure the SQL adapter to receive query notification messages from a SQL Server database. To demonstrate notifications, consider a table, Employee, with a "Status" column. When a new record is inserted to this table, the value of the Status column is set to 0. You can configure the adapter to receive notifications by registering for notifications using a SQL statement that retrieves all records that have Status column as "0." You can do so by specifying the SQL statement for the **NotificationStatement** binding property. After the adapter client receives the notification, it can contain the logic to do any subsequent tasks on the SQL Server database. In this example, for the sake of simplicity, the adapter client lists all the records in the table that have the Status column as "0."

Note

If you are performing operation on tables that have columns of user-defined types, make sure you refer to [Operations on Tables and Views with User-Defined Types](#) topic before you start developing your application.

Configuring Notifications with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure receiving notifications from SQL Server. You must specify these binding properties while running the .NET application to receive the notifications from a SQL Server database.

Binding Property	Description
InboundOperationType	Specifies the inbound operation that you want to perform. To receive notification messages, set this to Notification .
NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for query notifications. The adapter gets a notification message from SQL Server only when the result set for the specified SQL statement changes.
NotifyOnListenerStart	Specifies whether the adapter sends a notification to the adapter clients when the listener is started.

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to receive notifications from SQL Server, read further.

Configuring Notifications Using the WCF Service Model

To receive the notifications using the WCF service model, you must:

1. Generate a WCF service contract (interface) for the **Notification** operation from the metadata exposed by the adapter. To do this, you could use the Add Adapter Service Reference Plug-in.
2. Generate a WCF client for the **Select** operation on the Employee table. To do this, you could use the Add Adapter Service Reference Plug-in.
3. Implement a WCF service from this interface.
4. Host this WCF service using a service host (**System.ServiceModel.ServiceHost**).

About the Examples Used in this Topic

The examples in this topic receive notification for the Employee table. A script to generate the table is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **Notification_ServiceModel**, which is based on this topic, is also provided with the SQL adapter samples.

The WCF Service Contract and Class

You can use the Add Adapter Service Reference Plug-in to create a WCF service contract (interface) and supporting classes for

the **Notification** operation. For more information about generating a WCF service contract, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#).

The WCF Service Contract (Interface)

The following code shows the WCF service contract (interface) generated for the **Notification** operation.

```
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.ServiceContractAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/", ConfigurationName="NotificationOperation")]
public interface NotificationOperation {

    // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/Sql/2008/05/Notification/) of message
    // Notification does not match the default value (http://schemas.microsoft.com/Sql/2008/05/)
    [System.ServiceModel.OperationContractAttribute(IsOneWay=true, Action="Notification")]
    void Notification(Notification request);
}
```

The Message Contracts

Following is the message contract for the Notification operation.

```
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.CodeDom.Compiler.GeneratedCodeAttribute("System.ServiceModel", "3.0.0.0")]
[System.ServiceModel.MessageContractAttribute(WrapperName="Notification", WrapperNamespace="http://schemas.microsoft.com/Sql/2008/05/Notification/", IsWrapped=true)]
public partial class Notification {

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/Notification/", Order=0)]
    public string Info;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/Notification/", Order=1)]
    public string Source;

    [System.ServiceModel.MessageBodyMemberAttribute(Namespace="http://schemas.microsoft.com/Sql/2008/05/Notification/", Order=2)]
    public string Type;

    public Notification() {
    }

    public Notification(string Info, string Source, string Type) {
        this.Info = Info;
        this.Source = Source;
        this.Type = Type;
    }
}
```

WCF Service Class

The Add Adapter Service Reference Plug-in also generates a file that has a stub for the WCF service class implemented from the service contract (interface). The name of the file is `SqlAdapterBindingService.cs`. You can insert the logic to process the **Notification** operation directly into this class. The following code shows the WCF service class generated by the Add Adapter Service Reference Plug-in.

```
namespace SqlAdapterBindingNamespace {

    public class SqlAdapterBindingService : NotificationOperation {

        // CODEGEN: Generating message contract since the wrapper namespace (http://schemas.microsoft.com/Sql/2008/05/Notification/)
        // of message Notification does not match the default value (http://schemas.microso
```

```

ft.com/Sql/2008/05/)
    public virtual void Notification(Notification request) {
        throw new System.NotImplementedException("The method or operation is not implemented.");
    }
}
}
}

```

Receiving Query Notifications Using WCF Service Model

This section provides instructions on how to write a .NET application to receive query notifications using the SQL adapter.

To receive query notifications

1. Use the Add Adapter Service Reference Plug-in to generate a WCF client for **Select** operation on the **Employee** table. You will use this client to perform Select operations after receiving a notification message. Add a new class, TableOperation.cs to your project and add the following code snippet to perform a Select operation.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Notification_ServiceModel
{
    public class TableOperation
    {
        public void TableOp()
        {
            ////////////////////////////////////////////////////
            // CREATING THE CLIENT
            ////////////////////////////////////////////////////

            TableOp_dbo_EmployeeClient client = new TableOp_dbo_EmployeeClient("SqlAdapterBinding_TableOp_dbo_Employee");

            client.ClientCredentials.UserName.UserName = "<Enter user name here>";
            client.ClientCredentials.UserName.Password = "<Enter password here>";

            ////////////////////////////////////////////////////
            // OPENING THE CLIENT
            ////////////////////////////////////////////////////

            try
            {
                Console.WriteLine("Opening Client...");
                client.Open();
            }
            catch (Exception ex)
            {
                Console.WriteLine("Exception: " + ex.Message);
                throw;
            }

            ////////////////////////////////////////////////////
            // SELECTING THE LAST INSERTED RECORD FROM THE TABLE
            ////////////////////////////////////////////////////
            schemas.microsoft.com.Sql._2008._05.Types.Tables.dbo.Employee[] selectRecords;

```

```

try
{
    selectRecords = client.Select("*", "where Status=0");
}

catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}

Console.WriteLine("The details of the newly added employee are:");
Console.WriteLine("*****");
for (int i = 0; i < selectRecords.Length; i++)
{
    Console.WriteLine("Employee Name      : " + selectRecords[i].Name);
    Console.WriteLine("Employee Designation: " + selectRecords[i].Designat
ion);

    Console.WriteLine("Employee Status   : " + selectRecords[i].Status);
    Console.WriteLine();
}
Console.WriteLine("*****");

```

◆ Important

Because this code snippet performs operations on the Employee table that contains a Point UDT column, make sure you put the UDT DLL under the project's \bin\Debug folder while running the application.

2. Use the Add Adapter Service Reference Plug-in to generate a WCF service contract (interface) and helper classes for the **Notification** operation.

For more information, see [Generating a WCF Client or WCF Service Contract for SQL Server Artifacts](#). You can optionally specify the binding properties while generating the service contract and helper classes. This guarantees that they are properly set in the generated configuration file.

3. Implement a WCF service from the interface and helper classes generated in step 2. The **Notification** method of this class can throw an exception to abort the operation, if an error is encountered processing the data received from the **Notification** operation; otherwise the method does not return anything. You must attribute the WCF service class as follows:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
```

Within the **Notification** method, you can implement your application logic directly. This class can be found in `SqlAdapterBindingService.cs`. This code in this example sub-classes the **SqlAdapterBindingService** class. In this code, the notification message received is written to the console. Additionally, the **TableOp** method within the **TableOperation** class is invoked to perform the Select operation.

```

[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]

public class NotificationService : SqlAdapterBindingNamespace.SqlAdapterBindingService
{
    public override void Notification(Notification request)
    {
        Console.WriteLine("\nNew Notification Received");
        Console.WriteLine("*****");
        Console.WriteLine(request.Info);
        Console.WriteLine(request.Source);
        Console.WriteLine(request.Type);
    }
}

```

```

        Console.WriteLine("*****");

        // Invoke th TableOp method in the TableOperation class
        TableOperation Ops = new TableOperation();
        Ops.TableOp();
    }
}

```

4. Because the SQL adapter does not accept credentials as part of the connection URI, you must implement the following class to pass credentials for the SQL Server database. In the latter part of the application, you will instantiate this class to pass on the SQL Server credentials.

```

class NotificationCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bindingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase serviceHostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new NotificationCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

```

5. Create a **SqlAdapterBinding** and configure the adapter to receive query notifications by specifying the binding properties. You can do this either explicitly in code or declaratively in configuration. At a minimum, you must specify the **InboundOperationType** and **NotificationStatement** binding properties.

```

SqlAdapterBinding binding = new SqlAdapterBinding();
binding.InboundOperationType = InboundOperation.Notification;
binding.NotificationStatement = "SELECT Employee_ID, Name FROM dbo.Employee WHERE Status=0";
binding.NotifyOnListenerStart = true;

```

6. Specify SQL Server database credentials by instantiating the **NotificationCredentials** class you created in Step 4.

```

NotificationCredentials credentials = new NotificationCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

```

7. Create an instance of the WCF service created in step 3.

```

// create service instance

```

```
NotificationService service = new NotificationService();
```

8. Create an instance of **System.ServiceModel.ServiceHost** by using the WCF service and a base connection URI. You must also specify the credentials here.

```
// Enable service host
Uri[] baseUri = new Uri[] { new Uri("mssql://mysqlserver//mydatabase") };
ServiceHost serviceHost = new ServiceHost(service, baseUri);
serviceHost.Description.Behaviors.Add(credentials);
```

9. Add a service endpoint to the service host. To do this:

- Use the binding created in step 5.
- Specify a connection URI that contains credentials and, if required, an inbound ID.
- Specify the contract as "NotificationOperation".

```
// Add service endpoint: be sure to specify NotificationOperation as the contract
Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");
serviceHost.AddServiceEndpoint("NotificationOperation", binding, ConnectionUri);
```

10. To receive notification message, open the service host.

```
// Open the service host to begin receiving notifications
serviceHost.Open();
```

11. To stop receiving notifications, close the service host.

```
serviceHost.Close();
```

Example

The following example shows a .NET application to receive notification messages for the Employee table.

Note

The following code snippet instantiates a **TableOperation.cs** class and invokes the **TableOp** method. The class and the method are described in Step 1.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using Microsoft.Adapters.Sql;
using Microsoft.ServiceModel.Channels;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;
using System.Collections.ObjectModel;

namespace Notification_ServiceModel
{
    [ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
    public class NotificationService : SqlAdapterBindingNamespace.SqlAdapterBindingService
    {
```

```

public override void Notification(Notification request)
{
    Console.WriteLine("\nNew Notification Received");
    Console.WriteLine("*****");
    Console.WriteLine(request.Info);
    Console.WriteLine(request.Source);
    Console.WriteLine(request.Type);
    Console.WriteLine("*****");
    TableOperation Ops = new TableOperation();
    Ops.TableOp();
}
}

class NotificationCredentials : ClientCredentials, IServiceBehavior
{
    public void AddBindingParameters(ServiceDescription serviceDescription, ServiceHost
Base serviceHostBase, Collection<ServiceEndpoint> endpoints, BindingParameterCollection bin
dingParameters)
    {
        bindingParameters.Add(this);
    }

    public void ApplyDispatchBehavior(ServiceDescription serviceDescription, ServiceHos
tBase serviceHostBase)
    { }

    public void Validate(ServiceDescription serviceDescription, ServiceHostBase service
HostBase)
    { }

    protected override ClientCredentials CloneCore()
    {
        ClientCredentials clone = new NotificationCredentials();
        clone.UserName.UserName = this.UserName.UserName;
        clone.UserName.Password = this.UserName.Password;
        return clone;
    }
}

class Program
{
    static void Main(string[] args)
    {
        ServiceHost serviceHost = null;
        try
        {
            SqlAdapterBinding binding = new SqlAdapterBinding();
            binding.InboundOperationType = InboundOperation.Notification;
            binding.NotificationStatement = "SELECT Employee_ID, Name FROM dbo.Employee
WHERE Status=0";
            binding.NotifyOnListenerStart = true;

            // This URI is used to specify the address for the ServiceEndpoint
            // It must contain the InboundId (if any) that was used to generate
            // the WCF service callback interface
            Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");

            // This URI is used to initialize the ServiceHost. It cannot contain
            // a query_string (InboundID); otherwise, an exception is thrown when
            // the ServiceHost is initialized.
            Uri[] baseUri = new Uri[] { new Uri("mssql://mysqlserver//mydatabase") };

            NotificationCredentials credentials = new NotificationCredentials();
            credentials.UserName.UserName = "<Enter user name here>";
            credentials.UserName.Password = "<Enter password here>";

            Console.WriteLine("Opening service host...");
            NotificationService service = new NotificationService();

```


Developing Applications by Using the WCF Channel Model

You can use the Windows Communication Foundation (WCF) channel model to consume the Microsoft BizTalk Adapter for SQL Server by sending XML messages directly over a channel instance created with the SQL Server Binding.

One advantage of using the WCF channel model over using the strongly-typed classes and methods that the WCF service model exposes is that the channel model provides more fine-grained control over the operations that you perform on the SQL database. This control comes from the fact that in the WCF channel model, you directly control the contents of the messages that you send over the channel.

In certain scenarios, this extra level of control can be beneficial. For example, when you use the WCF channel model to perform an Update operation on a table, you can selectively update columns in the target rows by omitting columns from the update template that you pass in the message. The only columns that are required are those with "nillable=false" in the WSDL. The update method exposed by a WCF client uses a strongly-typed record parameter for the template that includes every column in the table schema.

The sections in this topic explain how to perform operations on the SQL adapter by using the WCF channel model.

In This Section

- [Overview of Using the WCF Channel Model with the Adapter](#)
- [Creating a Channel](#)
- [Performing an Insert Operation on a Table Using the WCF Channel Model](#)
- [Receiving Polling-based Data-changed Messages from SQL Server by Using the WCF Channel Model](#)

See Also

Other Resources

[Development](#)

Overview of Using the WCF Channel Model with the Adapter

To invoke operations on the Microsoft BizTalk Adapter for SQL Server, your code acts as a WCF client and sends outbound operations to the adapter. In the WCF channel model, your code invokes operations on the adapter by sending a request message over a channel.

To receive polling-based data-changed messages using the adapter, your code acts as a WCF service and receives the inbound **Polling**, **TypedPolling**, or **Notification** operation from the adapter. In other words, your code receives a request message for these operations from the adapter over a channel.

The topics in this section provide an overview of using the SQL adapter with the WCF channel model.

WCF Channel Model Overview

Clients and services communicate by exchanging SOAP messages. The WCF channel model is a low-level abstraction of this message exchange. It provides interfaces and types that enable you to send and receive messages by using a layered protocol stack called a channel stack. Each layer of the stack is composed of a channel, and each channel is created from a WCF binding. At the lowest layer is the transport channel. The transport channel implements the underlying transport mechanism between a service and a client and presents each message to the higher layers (and ultimately the consuming application) as a **System.ServiceModel.Message**. The WCF **Message** class is an abstraction of a SOAP message. WCF provides several channel interfaces, called channel shapes, that model the basic SOAP message exchange patterns, such as request-reply or one-way. A WCF transport binding provides an implementation of one or more channel shapes that higher layers can use to send and receive messages. For more information about the WCF channel model, see "Channel Model Overview" at <http://go.microsoft.com/fwlink/?LinkId=82614>.

The SQL adapter is a WCF custom transport binding that exposes a SQL Server database as a WCF service.

Supported Channel Shapes for the SQL Server Adapter

The adapter implements the following WCF channel shapes:

- **IRequestChannel (System.ServiceModel.Channels.IRequestChannel)**. The **IRequestChannel** interface implements the client side of a request-reply message exchange. You can use an **IRequestChannel** to perform operations for which you want to consume a response, for example to perform a SELECT query on a table.
- **IOutputChannel (System.ServiceModel.Channels.IOutputChannel)**. This shape implements the client side of a one-way message exchange. You can use an **IOutputChannel** to invoke an operation for which you do not need to consume a response, for example to call a procedure that has no return parameters.

◆ Important

All underlying calls by the adapter to the SQL Server client are synchronous. This includes calls to the SQL Server client that are the result of operations invoked over an **IOutputChannel**. When you use an **IOutputChannel**, the adapter discards the response received from the SQL Server client.

- **IInputChannel (System.ServiceModel.Channels.IInputChannel)**. This shape implements the service side of a one-way message exchange. You use an **IInputChannel** to receive messages for inbound operations, such as **Polling** or **Notification**, from the adapter.

Like any WCF binding, the SQL adapter uses a factory pattern to provide channels to application code. You use a **Microsoft.Adapters.SQLBinding** object to create instances of:

- **System.ServiceModel.ChannelFactory<IRequestChannel>** to provide **IRequestChannel** channels you can use to invoke request-response operations on the adapter.
- **System.ServiceModel.ChannelFactory<IOutputChannel>** to provide **IOutputChannel** channels you can use to invoke one-way operations on the adapter.
- **System.ServiceModel.IChannelListener<IInputChannel>** to provide **IInputChannel** channels you can use to receive messages for inbound operations, such as **Polling** or **Notification**, from the adapter.

Creating Messages for the SQL Server Database Adapter in the WCF Channel Model

In WCF the **System.ServiceModel.Channels.Message** class provides an in memory representation of a SOAP message. You create a **Message** instance by invoking the static **Message.Create** method.

There are two important parts to the SOAP message that you must specify when you create a **Message** instance to send to the SQL adapter.

- The message action is a string that is part of the SOAP message header. The message action identifies the operation that should be invoked on the database. The following shows the message action specified to invoke the Select operation on the Employee table: `TableOp/Select/dbo/Employee`.
- The message body contains the parameter data for the operation. The message body is composed of well-formed XML that corresponds to the message schema expected by the SQL adapter for the requested operation. The following message body specifies a Select operation on the Employee table (`SELECT * FROM Employee WHERE Employee_ID=10001`).

```
<Select xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <Columns>*</Columns>
  <Query>where Employee_ID=10001</Query>
</Select>
```

For information about the SQL adapter message schemas and message actions for operations, see [Messages and Message Schemas for BizTalk Adapter for SQL Server](#).

This **Create** method is overloaded and offers many different options for providing the message body. The following code shows how to create a **Message** instance by using an **XmlReader** to supply the message body. In this code, the message body is read from a file.

```
XmlReader readerIn = XmlReader.Create("SelectInput.xml");
Message messageIn = Message.CreateMessage(MessageVersion.Default,
    "TableOp/Select/dbo/Employee",
    readerIn);
```

◆ Important

You must provide a message action in your **Message** instance. This is typically done when the **Message** instance is created.

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Creating a Channel

In the WCF channel model, you invoke operations on the SQL Server database and receive the results by exchanging SOAP messages with the Microsoft BizTalk Adapter for SQL Server over a WCF channel.

- You invoke outbound operations by using either an **IRequestChannel** or an **IOutputChannel** to send messages to the adapter.
- You receive messages for inbound operations by receiving messages over an **IInputChannel** for **Polling**, **TypedPolling**, or **Notification** operations.

The procedures in this topic provide information about how to create and configure channel shapes that are used for inbound and outbound operations.

Creating Outbound (Client) Channels

You can use either an **IRequestChannel** or an **IOutputChannel** to invoke operations on the SQL Server database. In either case, you first create a **System.ServiceModel.ChannelFactory** using the appropriate interface. You then use the factory to create the channel. After you have created the channel you can use it to invoke operations on the adapter.

To create and open an outbound channel

1. Create and initialize an instance of **ChannelFactory** for the desired channel shape by using an endpoint and a binding. The endpoint specifies a SQL Server connection URI and the binding is an instance of **sqlBinding**.
2. Provide SQL Server credentials for the channel factory by using the **Credentials** property.
3. Open the channel factory.
4. Get an instance of the channel by invoking the **CreateChannel** method on the channel factory.
5. Open the channel.

You can specify the binding and endpoint address in your code or from configuration.

Specifying the Binding and Endpoint Address in Code

The following code example shows how to create an **IRequestChannel** by specifying the binding and endpoint address in code. The code to create an **IOutputChannel** is the same except that you must specify an **IOutputChannel** interface for the **ChannelFactory** and channel type.

```
// Create binding -- set binding properties before you open the factory.
SqlAdapterBinding sdbBinding = new SqlAdapterBinding();

// Create address.
EndpointAddress sdbAddress = new EndpointAddress("mssql://<sql_server_name>//<database_name>?");

// Create channel factory from binding and address.
ChannelFactory<IRequestChannel> factory =
    new ChannelFactory<IRequestChannel>(sdbBinding, sdbAddress);

// Specify credentials.
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open factory
factory.Open();

// Get channel and open it.
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

Specifying the Binding and Endpoint Address in Configuration

The following code example shows how to create a channel factory from a client endpoint specified in configuration.

```
// Create channel factory from configuration.
ChannelFactory<IRequestChannel> factory =
new ChannelFactory<IRequestChannel>("MyRequestChannel");

// Specify credentials.
factory.Credentials.UserName.UserName = "myuser";
factory.Credentials.UserName.Password = "mypassword";

// Open the factory.
factory.Open();

// Get a channel and open it.
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

The Configuration Settings

The following code shows the configuration settings used for the preceding example. The contract for the client endpoint must be "System.ServiceModel.Channels.IRequestChannel" or "System.ServiceModel.Channels.IOutputChannel" depending on the kind of channel shape that you want to create.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">
  <system.serviceModel>
    <bindings>
      <sqlBinding>
        <binding name="SqlAdapterBinding" closeTimeout="00:01:00" openTimeout="00:0
1:00"
          receiveTimeout="00:10:00" sendTimeout="00:01:00" maxConnectionPoolSize=
"100"
          encrypt="false" useAmbientTransaction="true" batchSize="20"
          polledDataAvailableStatement="" pollingStatement="" pollingIntervalInSe
conds="30"
          pollWhileDataFound="false" notificationStatement="" notifyOnListenerSta
rt="true"
          enableBizTalkCompatibilityMode="true" chunkSize="4194304"
          inboundOperationType="Polling" useDatabaseNameInXsdNamespace="false"
          allowIdentityInsert="false" enablePerformanceCounters="false"
          xmlStoredProcedureRootNodeName="" xmlStoredProcedureRootNodeNamespace="
" />
      </sqlBinding>
    </bindings>
    <client>
      <endpoint address="mssql://mysqlserver//mydatabase?" binding="sqlBinding"
        bindingConfiguration="SqlAdapterBinding" contract="System.ServiceModel.Chan
nels.IRequestChannel"
        name="MyRequestChannel" />
    </client>
  </system.serviceModel>
</configuration>
```

Creating Inbound (Service) Channels

You configure the SQL adapter to poll the SQL Server database tables and views by setting binding properties on an instance of **sqlBinding**. You then use this binding to build a channel listener from which you can get an **IInputChannel** channel to receive the **Polling**, **TypedPolling**, or **Notification** operation from the adapter.

To create and open an IInputChannel to receive inbound operations

1. Create an instance of **SQLBinding**.
2. Set the binding properties required for inbound operation. For example, for a **Polling** operation, at a minimum you must set the **InboundOperationType**, **PolledDataAvailableStatement**, and **PollingStatement** binding properties to configure the SQL adapter to poll the SQL Server database.

3. Create a channel listener by invoking **BuildChannelListener<IInputChannel>** method on the **SQLBinding**. You specify the SQL Server connection URI as one of the parameters to this method.
4. Open the listener.
5. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on listener.
6. Open the channel.

The following code shows how to create a channel listener and get an **IInputChannel** to receive data-changed messages from the adapter.

◆ Important

The SQL adapter only supports one-way receive. So, you must use **IInputChannel** to receive messages for inbound operations from SQL Server.

```
// Create a binding: specify the InboundOperationType, the PolledDataAvailableStatement, and
// the PollingStatement binding properties.
SqlAdapterBinding binding = new SqlAdapterBinding();
binding.InboundOperationType = InboundOperation.Polling;
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";
binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS
John, Tester, 100000";

// Create a binding parameter collection and set the credentials
ClientCredentials credentials = new ClientCredentials();
credentials.UserName.UserName = "myuser";
credentials.UserName.Password = "mypassword";

BindingParameterCollection bindingParams = new BindingParameterCollection();
bindingParams.Add(credentials);

// Get a listener from the binding and open it.
Uri connectionUri = new Uri("mssql://mysqlserver//mydatabase?");
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>(conn
ectionUri, bindingParams);
listener.Open();

// Get a channel from the listener and open it.
IInputChannel channel = listener.AcceptChannel();
channel.Open();
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Performing an Insert Operation on a Table Using the WCF Channel Model

The SQL adapter discovers a set of basic Insert, Select, Update, and Delete operations on SQL Server database tables and views. By using these operations, you can perform simple SQL Insert, Select, Update, and Delete statements qualified by a Where clause on a target table or view. This topic provides instructions on how to perform an Insert operation on a SQL Server database table using the WCF channel model.

For more information on how the adapter supports these operations, see [Performing Basic Insert, Update, Delete, and Select Operations on Tables and Views](#). For more information about how to perform operations on SQL Server using the WCF Channel model, see [Overview of Using the WCF Channel Model with the Adapter](#).

About the Examples Used in this Topic

The example in this topic performs operations on the Employee table. The Employee table is created by running the SQL script provided with the samples. For more information about samples, see [Samples](#). A sample, **EmployeeInsertOp**, which is based on this topic, is also provided with the SQL adapter samples.

The Insert Message

To perform operations on the SQL Server database using the WCF channel model, you must have the request message specific to the operation. The request message to perform an Insert operation on the Employee table in the SQL Server database resembles the following:

```
<Insert xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">
  <Rows>
    <Employee xmlns="http://schemas.microsoft.com/Sql/2008/05/Types/Tables/dbo">
      <Name>Tom Smith</Name>
      <Designation>Manager</Designation>
      <Salary>500000</Salary>
    </Employee>
  </Rows>
</Insert>
```

This request message inserts a record with following details:

```
Name = Tom Smith
Designation = Manager
Salary = 500000
```

You must copy the message to a file, e.g. InsertRequest.xml. This file is used in this example to send the request message to SQL Server using the SQL adapter. For more information about the message schema for operations on table, see [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#).

Creating a WCF Channel Application

This section provides instructions on how to create a WCF channel application to perform an Insert operation on the Employee table.

To create a WCF channel application for inserting records into the Employee table

1. Create a Visual C# project in Visual Studio. For this topic, create a console application.
2. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql`, `Microsoft.ServiceModel.Channels`, `System.ServiceModel`, and `System.Runtime.Serialization`.
3. Open the Program.cs file and add the following namespaces:
 - `Microsoft.Adapters.Sql`
 - `Microsoft.ServiceModel.Channels`

- System.ServiceModel
- System.ServiceModel.Channels
- System.Xml

4. Create the binding and endpoint.

```
SqlAdapterBinding binding = new SqlAdapterBinding();
EndpointAddress address = new EndpointAddress("mssql://mysqlserver//mydatabase?");
```

5. Create and open the channel factory. This application sends request message to SQL Server and receives a response, hence you must implement the IRequestChannel interface.

```
ChannelFactory<IRequestChannel> factory = new ChannelFactory<IRequestChannel>(binding,
address);
factory.Credentials.UserName.UserName = "<Enter user name here>";
factory.Credentials.UserName.Password = "<Enter password here>";
factory.Open();
```

6. Create and open the channel.

```
IRequestChannel channel = factory.CreateChannel();
channel.Open();
```

7. Create and send the request message.

```
XmlReader readerIn;
Console.WriteLine("Creating the message");
try
{
    readerIn = XmlReader.Create("InsertRequest.xml");
    Console.WriteLine("Reader created");
}
catch (Exception ex)
{
    Console.WriteLine("Exception: " + ex.Message);
    throw;
}
Message messageIn = Message.CreateMessage(MessageVersion.Default, "TableOp/Insert/dbo/
Employee", readerIn);
Message messageOut = channel.Request(messageIn);
```

While creating the request message, you must specify the message action that indicates the action that the adapter performs on the SQL Server table. To perform an Insert operation on the Employee table, the message action is `TableOp/Insert/dbo/Employee`. For information about how you can determine the message action for various operations on tables, see [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#).

8. Get the response message.

```
XmlReader readerOut = messageOut.GetReaderAtBodyContents();
XmlDocument doc = new XmlDocument();
doc.Load(readerOut);
doc.Save("C:\\Response.xml");
```

9. Close the message, channel, and channel factory.

```
messageOut.Close();  
channel.Close();  
factory.Close();
```

10. Build the project. After building the project, you must perform the following tasks:

- Copy the request message, InsertRequest.xml, at the same location as your project executable. Typically, this location is \bin\Debug\ under your project directory.
- The "Employee" table used in this example has a column of Point user-defined type (UDT). So, before running the project you must create the assembly for the Point UDT as described at <http://go.microsoft.com/fwlink/?LinkID=138066>. You must also copy the assembly DLL at the same location as the project executable. Typically, this location is \bin\Debug\ under your project directory.

11. Run the application. The response message, Response.xml, is saved at the location you specified in the application. The response message contains the ID of the newly added employee and resembles the following:

```
<InsertResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/dbo/Employee">  
  <InsertResult>  
    <long xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays">10006</long>  
  </InsertResult>  
</InsertResponse>
```

Because the Employee table has the Employee_ID column as the identity column, the Insert operation returns the value for the identity column of the newly inserted record. If there is no identity column in a table, the return value is NULL.

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Receiving Polling-based Data-changed Messages from SQL Server by Using the WCF Channel Model

You can configure the SQL adapter to receive periodic data-change messages for SQL Server tables or views. You can specify a polling statement that the adapter executes to poll the database. The polling statement can be a SELECT statement or a stored procedure that returns a result set.

For more information on how the adapter supports polling, see [Support for Polling](#).

◆ Important

If you want to have more than one polling operation in a single application, you must specify an **InboundID** connection property as part of the connection URI to make it unique. The inbound ID you specify is added to the operation namespace to make it unique.

How This Topic Demonstrates Polling

In this topic, to demonstrate how the SQL adapter supports receiving data change messages, create a .NET application for the **Polling** operation. For this topic, specify the **PolledDataAvailableStatement** as:

```
SELECT COUNT(*) FROM Employee
```

The **PolledDataAvailableStatement** must return a result set with the first cell containing a positive value. If the first cell does not contain a positive value, the adapter does not execute the polling statement.

As part of the polling statement, perform the following operations:

1. Select all the rows from the Employee table.
2. Execute a stored procedure (MOVE_EMP_DATA) to move all the records from the Employee table to an EmployeeHistory table.
3. Execute a stored procedure (ADD_EMP_DETAILS) to add a new record to the Employee table. This procedure takes the employee name, designation, and salary as parameters.

To perform these operations, you must specify the following for the **PollingStatement** binding property:

```
SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000
```

After the polling statement is executed, all the records from the Employee table are selected and the message from SQL Server is received. Once the MOVE_EMP_DATA stored procedure is executed by the adapter, all the records are moved to the EmployeeHistory table. Then, the ADD_EMP_DETAILS stored procedure is executed to add a new record to the Employee table. The next polling execution will only return a single record. This cycle continues until you close the channel listener.

Configuring a Polling Query with the SQL Adapter Binding Properties

The following table summarizes the SQL adapter binding properties that you use to configure the adapter to receive data-change messages. You must specify these binding properties as part of the .NET application for polling.

Binding Property	Description
InboundOperationType	Specifies whether you want to perform Polling , TypedPolling , or Notification inbound operation. Default is Polling .

PolledDataAvailableStatement	Specifies the SQL statement that the adapter executes to determine whether any data is available for polling. The SQL statement must return a result set consisting of rows and columns. Only if a row is available, the SQL statement specified for the PollingStatement binding property will be executed.
PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property. The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter waits for the remaining time in the interval.
PollingStatement	Specifies the SQL statement to poll the SQL Server database table. You can specify a simple SELECT statement or a stored procedure for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property. You can specify any number of SQL statements separated by a semi-colon.
PollWhenDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled. If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval. Default is false .

For a more complete description of these properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For a complete description of how to use the SQL adapter to poll SQL Server, read the remainder of this topic.

Consuming the Polling Request Message

The adapter invokes the **Polling** operation on your code to poll the SQL Server database. That is, the adapter sends a Polling request message that you receive over an `IInputChannel` channel shape. The Polling request message contains the result set of the query specified by the `PollingStatement` binding property. You can consume the Polling message in one of two ways:

- To consume the message using node-value streaming you must call the **WriteBodyContents** method on the response message and pass it an **XmlDictionaryWriter** that implements node-value streaming.
- To consume the message using node streaming you can call **GetReaderAtBodyContents** on the response message to get an **XmlReader**.

About the Examples Used in this Topic

The examples in this topic poll the Employee table. The example also uses the `MOVE_EMP_DATA` and `ADD_EMP_DETAILS` stored procedure. A script to generate these artifacts is supplied with the samples. For more information about the samples, see [Samples](#). A sample, **Polling_ChannelModel**, which is based on this topic, is also provided with the SQL adapter samples.

Receiving Inbound Messages for Polling Operation Using the WCF Channel Model

This section provides instructions on how to write a .NET application (channel model) to receive inbound polling messages using the SQL adapter.

To receive polling messages from the SQL adapter

1. Create a Microsoft Visual C# project in Visual Studio. For this topic, create a console application.
2. In the Solution Explorer, add reference to `Microsoft.Adapters.Sql`, `Microsoft.ServiceModel.Channels`, `System.ServiceModel`, and `System.Runtime.Serialization`.
3. Open the Program.cs file and add the following namespaces:
 - `Microsoft.Adapters.Sql`
 - `System.ServiceModel`
 - `System.ServiceModel.Description`

- `System.ServiceModel.Channels`
- `System.Xml`

4. Specify a connection URI. For more information about the adapter connection URI, see [SQL Server Connection URI](#).

```
Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");
```

5. Create an instance of **SqlAdapterBinding** and set the binding properties required to configure polling. At a minimum you must set the **InboundOperationType**, **PolledDataAvailableStatement**, and **PollingStatement** binding properties. For more information about binding properties used to configure polling, see [Support for Polling](#).

```
SqlAdapterBinding binding = new SqlAdapterBinding();  
binding.InboundOperationType = InboundOperation.Polling;  
binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";  
binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DET  
AILS John, Tester, 100000";
```

6. Create a binding parameter collection and set the credentials.

```
ClientCredentials credentials = new ClientCredentials();  
credentials.UserName.UserName = "<Enter user name here>";  
credentials.UserName.Password = "<Enter password here>";  
  
BindingParameterCollection bindingParams = new BindingParameterCollection();  
bindingParams.Add(credentials);
```

7. Create a channel listener and open it. You create the listener by invoking **BuildChannelListener<IInputChannel>** method on the **SqlAdapterBinding**.

```
IChannelListener<IInputChannel> listener = binding.BuildChannelListener<IInputChannel>  
(connectionUri, bindingParams);  
listener.Open();
```

8. Get an **IInputChannel** channel by invoking the **AcceptChannel** method on the listener and open it.

```
IInputChannel channel = listener.AcceptChannel();  
channel.Open();
```

9. Invoke **Receive** on the channel to get the next POLLINGSTMT message from the adapter.

```
Message message = channel.Receive();
```

10. Consume the result set returned by the POLLINGSTMT operation. You can consume the message using either an **XmlReader** or an **XmlDictionaryWriter**.

```
XmlReader reader = message.GetReaderAtBodyContents();
```

11. Close the channel when you have completed processing the request.

```
channel.Close();
```

◆ Important

You must close the channel after you have finished processing the POLLINGSTMT operation. Failure to close the channel may affect the behavior of your code.

12. Close the listener when you are finished receiving data-changed messages.

```
listener.Close()
```

◆ Important

Closing the listener does not close channels created using the listener. You must explicitly close each channel created using the listener.

Example

The following example shows a polling query that executes the Employee table. The polling statement performs the following tasks:

- Selects all the records from the Employee table.
- Executes the MOVE_EMP_DATA stored procedure to move all records from Employee table to EmployeeHistory table.
- Executes the ADD_EMP_DETAILS stored procedure to add a single record to the Employee table.

The polling messages are saved at C:\PollingOutput.xml.

```
using System;
using Microsoft.Adapters.Sql;
using System.ServiceModel;
using System.ServiceModel.Description;
using System.ServiceModel.Channels;

using System.Xml;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Sample started. This sample will poll 5 times and will perform the following tasks:");
            Console.WriteLine("Press any key to start polling...");
            Console.ReadLine();
            IChannelListener<IInputChannel> listener = null;

            IInputChannel channel = null;

            try
            {
                TimeSpan messageTimeout = new TimeSpan(0, 0, 30);

                SqlAdapterBinding binding = new SqlAdapterBinding();
                binding.InboundOperationType = InboundOperation.Polling;
                binding.PolledDataAvailableStatement = "SELECT COUNT (*) FROM EMPLOYEE";
                binding.PollingStatement = "SELECT * FROM Employee;EXEC MOVE_EMP_DATA;EXEC ADD_EMP_DETAILS John, Tester, 100000";

                Uri ConnectionUri = new Uri("mssql://mysqlserver//mydatabase?");
```

```

ClientCredentials credentials = new ClientCredentials();
credentials.UserName.UserName = "<Enter user name here>";
credentials.UserName.Password = "<Enter password here>";

BindingParameterCollection bindingParams = new BindingParameterCollection()
;
bindingParams.Add(credentials);

listener = binding.BuildChannelListener<IInputChannel>(ConnectionUri, bindi
ngParams);
listener.Open();

channel = listener.AcceptChannel();
channel.Open();

Console.WriteLine("Channel and Listener opened...");
Console.WriteLine("\nWaiting for polled data...");
Console.WriteLine("Receive request timeout is {0}", messageTimeout);

// Poll five times with the specified message timeout
// If a timeout occurs polling will be aborted
for (int i = 0; i < 5; i++)
{
    Console.WriteLine("Polling: " + i);
    Message message = null;
    XmlReader reader = null;
    try
    {
        //Message is received so process the results
        message = channel.Receive(messageTimeout);
    }
    catch (System.TimeoutException toEx)
    {
        Console.WriteLine("\nNo data for request number {0}: {1}", i + 1, t
oEx.Message);
        continue;
    }

    // Get the query results using an XML reader
    try
    {
        reader = message.GetReaderAtBodyContents();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception :" + ex);
        throw;
    }

    XmlDocument doc = new XmlDocument();
    doc.Load(reader);
    using (XmlWriter writer = XmlWriter.Create("C:\\PollingOutput.xml"))
    {
        doc.WriteTo(writer);
        Console.WriteLine("The polling response is saved at 'C:\\PollingOut
put.xml'");
    }

    // return the cursor
    Console.WriteLine();

    // close the reader
    reader.Close();

    message.Close();
}
Console.WriteLine("\nPolling done -- hit <RETURN> to finish");

```

```
        Console.ReadLine();
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception is: " + ex.Message);
        if (ex.InnerException != null)
        {
            Console.WriteLine("Inner Exception is: " + ex.InnerException.Message);
        }
    }
    finally
    {
        // IMPORTANT: close the channel and listener to stop polling
        if (channel != null)
        {
            if (channel.State == CommunicationState.Opened)
                channel.Close();
            else
                channel.Abort();
        }

        if (listener != null)
        {
            if (listener.State == CommunicationState.Opened)
                listener.Close();
            else
                listener.Abort();
        }
    }
}
}
```

See Also

Other Resources

[Developing Applications by Using the WCF Channel Model](#)

Samples

Samples for Microsoft BizTalk Adapter for SQL Server are categorized into:

- BizTalk Server samples
- WCF service model samples
- WCF channel model samples

The samples are available at <http://go.microsoft.com/fwlink/?LinkID=196854>. The SQL scripts for creating the objects used in the samples, such as database, tables, procedures, etc., are also available along with the samples.

The following list contains the names and descriptions of the samples for the SQL adapter.

BizTalk Server Samples

Sample Directory Name	Description
ExecuteStoredProcedure	Demonstrates how to invoke a stored procedure in SQL Server database using the adapter with BizTalk Server .
SelectTable	Demonstrates how to perform a Select operation on a SQL Server database table using the adapter with BizTalk Server.
CompositeOperations	Demonstrates how to perform composite operations on a SQL Server database using the adapter with BizTalk Server.
TypedPolling	Demonstrates how to perform strongly-typed polling on a SQL Server database using the adapter with BizTalk Server.
FILESTREAMOperation	Demonstrates how to perform FILESTREAM operations on a SQL Server 2008 database using the adapter with BizTalk Server.
IncrementalNotification	Demonstrates how to receive incremental notification from a SQL Server database using the adapter with BizTalk Server.
Employee_PurchaseOrder	Sample based on Tutorial 2: Employee - Purchase Order Process .

WCF Service Model Samples

Sample Directory Name	Description
EmployeeBasicOps	Demonstrates how to perform Insert, Select, Update, and Delete operation on a SQL Server database using the adapter.
ExecuteReader	Demonstrates how to invoke an ExecuteReader operation on a SQL Server database using the adapter.
Execute_StoredProc	Demonstrates how to invoke a stored procedure in SQL Server database using the adapter.
Execute_TypedStoredProcedure	Demonstrates how to invoke a strongly-typed stored procedure in SQL Server database using the adapter.
Records_FILESTREAM_Op	Demonstrates how to update FILESTREAM data in a SQL Server database table using the adapter.

ScalarFunction_ServiceModel	Demonstrates how to invoke scalar functions in a SQL Server database using the adapter.
TableFunction_ServiceModel	Demonstrates how to invoke table-valued functions in a SQL Server database using the adapter.
Polling_ServiceModel	Demonstrates how to receive polling-based data-changed messages from a SQL Server database using the adapter.
TypedPolling_ServiceModel	Demonstrates how to receive strongly-typed polling-based data-changed messages from a SQL Server database using the adapter.
Notification_ServiceModel	Demonstrates how to receive query notifications from a SQL Server database using the adapter.

WCF Channel Model Samples

Sample Directory Name	Description
EmployeeInsertOp	Demonstrates how to perform an Insert operation on a SQL Server database using the adapter.
Polling_ChannelModel	Demonstrates how to receive polling-based data-changed messages from a SQL Server database using the adapter.

See Also

Other Resources

[Development](#)

How Do I?

The topic in this section describes how to configure transaction isolation level and the transaction timeout value in BizTalk Server while using the Polling and Notification operations in the Microsoft BizTalk Adapter for SQL Server.

In This Section

[Configure Transaction Isolation Level and Transaction Timeout](#)

See Also

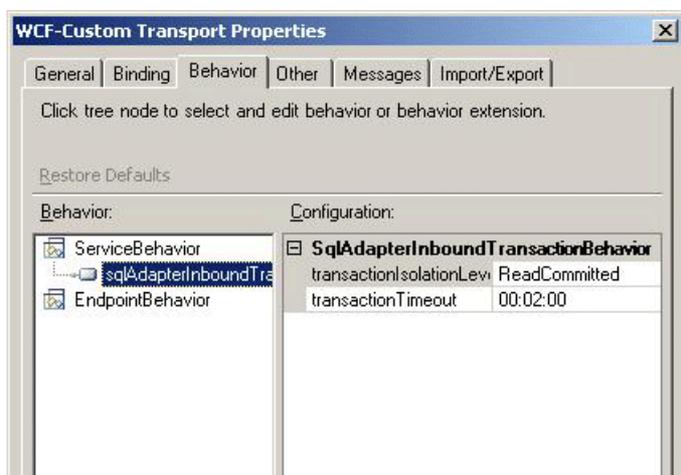
Other Resources

[Development](#)

Configure Transaction Isolation Level and Transaction Timeout

While performing inbound operations (Polling and Notification) using the Microsoft BizTalk Adapter for SQL Server with BizTalk Server, you should appropriately configure the transaction isolation level and the transaction timeout values. To do this:

1. Start the BizTalk Server Administration console.
2. In the console tree, expand the **BizTalk Group**, and then expand **Applications**.
3. Expand the application under which you want to deploy the SQL adapter.
4. Right-click **Receive Ports**, point to **New**, and then click **One-way Receive Port**.
5. In the **Receive Port Properties** dialog box, on the **General** tab, type a name for the receive port.
6. In the left pane of the **Receive Port Properties** dialog box, click **Receive Locations**, and then click **New** in the right pane to define a new receive location.
7. In the **Receive Location Properties** dialog box, click **WCF-Custom** in the **Type** list.
8. Click **Configure** adjacent to the **Type** list.
9. In the **WCF-Custom Transport Properties** dialog box, click the **Behavior** tab.
10. In the **Behavior** list, right-click **ServiceBehavior**, and click **Add extension**.
11. In the **Select Behavior Extension** dialog box, select **sqlAdapterInboundTransactionBehavior**, and click **OK**.
12. In the left pane of the **WCF-Custom Transport Properties**, select the **sqlAdapterInboundTransactionBehavior** service under **ServiceBehavior**.
13. In the right pane of the **WCF-Custom Transport Properties**, specify appropriate values for the **transactionIsolationLevel** and **transactionTimeout** parameters. You can select any of the following transaction isolation levels: **Serializable**, **RepeatableRead**, **ReadCommitted**, **ReadUncommitted**, **Snapshot**, **Chaos**, and **Unspecified**. For information about these transaction isolation levels, see the **Members** section at <http://go.microsoft.com/fwlink/?LinkId=126983>.



14. Click **OK** in the **WCF-Custom Transport Properties** dialog box.
15. Click **OK** in the open dialog boxes to save the changes.

Deployment

This section provides information about deploying the Microsoft BizTalk Adapter Pack. It also provides answers to some frequently asked questions and a link to information about troubleshooting common installation issues.

In This Section

- [Installing the BizTalk Adapter Pack](#)
- [Troubleshooting Your Installation](#)

Installing the BizTalk Adapter Pack

You can install the SQL adapter either as part of the BizTalk Adapter Pack installation or as a separate adapter.

- If you want to install the SQL adapter as part of the BizTalk Adapter Pack, see the BizTalk Adapter Pack installation guide (InstallationGuide.htm) for installation prerequisites and complete instructions for installing the BizTalk Adapter Pack. This document is installed on your computer, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter Pack\Documents.
- If you want to install only the SQL adapter as a separate adapter, see the SQL adapter installation guide (SQLAdapterInstallGuide.htm) for installation prerequisites and complete instructions for installing the SQL adapter. This document is installed on your computer, typically under <installation drive>:\Program Files\Microsoft BizTalk Adapter for SQL Server\Documents.

◆ Important

The BizTalk Adapter Pack and the SQL adapter that is shipped separately cannot coexist on the same computer.

See Also

Concepts

[Troubleshooting Your Installation](#)

Other Resources

[Deployment](#)

Troubleshooting Your Installation

Installation of the Microsoft BizTalk Adapter Pack or the separate SQL adapter copies the product binaries on a computer and registers the bindings for each adapter. For information about troubleshooting installation issues, see [Troubleshooting Installation Issues](#).

See Also

Tasks

[Troubleshooting Installation Issues](#)

Other Resources

[Deployment](#)

Troubleshooting

The BizTalk Adapter Pack uses or depends on several Microsoft technologies, including but not limited to the Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK, Microsoft Visual Studio, the Microsoft .NET Framework 3.5 SP1, and .NET Framework 4. The BizTalk Adapter Pack is built on top of the WCF LOB Adapter SDK, which in turn requires the .NET Framework 3.5 SP1 and .NET Framework 4. The adapters can be consumed either by writing applications using the Visual Studio or by creating BizTalk applications. For issues related to each of these technologies and products, see the respective documentation.

This section provides information about troubleshooting the BizTalk Adapter Pack, including:

- Enabling tracing to diagnose issues with the adapters.
- Handling installation and operational issues that you might encounter when working with the adapters, including probable cause, and a resolution.
- Using performance counters to gauge adapter performance.

-

In This Section

- [Diagnostic Tracing and Message Logging](#)
- [Troubleshooting Installation Issues](#)
- [Troubleshooting Operational Issues](#)
- [Using Performance Counters](#)

-

Diagnostic Tracing and Message Logging

Diagnostic tracing helps to effectively diagnose problems that you might encounter when using the adapters. Adapter clients can activate diagnostic tracing at two levels:

- Between the adapter client and the adapter
- Within the adapter

This section provides information about activating tracing at these levels.

Tracing Between the Adapter Client and the Adapter

Adapter clients can enable WCF tracing to trace issues between the adapter client and the adapter. WCF tracing is used to trace the input XML that comes from the adapter client by using the WCF service model and is useful in diagnosing serialization issues. WCF tracing is not used for the WCF channel model or for output messages from the adapter to the adapter client. You can activate WCF tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design time and run time.

- **Tracing at design time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in `<installation drive>:\Program Files\Microsoft Visual Studio <version>\Common7\IDE`.
- **Tracing at run time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server application, you must add the excerpt to the BizTalk configuration file, typically `BTSNTSvc.exe.config`. For BizTalk Server 2010, this file is available typically under `<installation drive>:\Program Files\Microsoft BizTalk Server 2010`.
 - For a WCF service model .NET application, you must add the excerpt to the `app.config` file of your project.

To enable WCF tracing, add the following excerpt within the `<configuration>` tag.

```
<system.diagnostics>
  <sources>
    <source name="System.ServiceModel" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.ServiceModel.MessageLogging"
      switchValue="Verbose, ActivityTracing">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="System.Runtime.Serialization" switchValue="Verbose">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\WCFTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>
<system.serviceModel>
```

```

<diagnostics>
  <messageLogging
    logEntireMessage="true"
    logMalformedMessages="false"
    logMessagesAtServiceLevel="true"
    logMessagesAtTransportLevel="false"/>
</diagnostics>
</system.serviceModel>

```

This saves the WCF traces to C:\log\WCFTrace.svclog. For more information about WCF tracing, see "Tracing" at <http://go.microsoft.com/fwlink/?LinkId=91242>.

◆ Important

Make sure you mitigate potential security threats of exposing sensitive business data by enabling tracing. For recommendations see [Best Practices](#).

Tracing Within the Adapter

The adapters log different categories of useful information to the trace file such as errors, warnings, and information messages. Such information is useful in understanding the process flow within the adapter and diagnosing issues with the adapter. You can activate the WCF LOB Adapter SDK and adapter tracing for BizTalk applications and WCF service model applications by adding an excerpt to the respective configuration files. Also, you can enable tracing both at design time and run time.

- **Tracing at design time.** For the design-time experience, you may use the Add Adapter Service Reference Plug-in, Consume Adapter Service Add-in, or the Add Adapter Metadata Wizard. All these tools can be used from Visual Studio. So, to enable tracing for the design-time experience, you must add the excerpt to the devenv.exe.config file located in <installation drive>:\Program Files\Microsoft Visual Studio <version>\Common7\IDE.
- **Tracing at run time.** For run-time tracing, you must add the excerpt depending on the application you are using.
 - For a BizTalk Server application, you must add the excerpt to the BizTalk configuration file, typically BTSNTSvc.exe.config. For BizTalk Server 2010, this file is available typically under <installation drive>:\Program Files\Microsoft BizTalk Server 2010.
 - For a WCF service model .NET application, you must add the excerpt to the app.config file of your project.

To enable WCF LOB Adapter SDK and adapter tracing, add the following excerpt within the <configuration> tag.

```

<system.diagnostics>
  <sources>
    <source name="Microsoft.ServiceModel.Channels" switchValue="Error">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
    <source name="Microsoft.Adapters.Sql" switchValue="Information">
      <listeners>
        <add name="xml" />
      </listeners>
    </source>
  </sources>
  <sharedListeners>
    <add name="xml" type="System.Diagnostics.XmlWriterTraceListener"
      traceOutputOptions="LogicalOperationStack"
      initializeData="C:\log\AdapterTrace.svclog" />
  </sharedListeners>
  <trace autoflush="true" />
</system.diagnostics>

```

This saves the WCF traces to C:\log\AdapterTrace.svclog.

Viewing the Traces

You can use the Windows Communication Foundation (WCF) Service Trace Viewer tool to view the traces. For more information about the tool, see "Using Service Trace Viewer for Viewing Correlated Traces and Troubles" at <http://go.microsoft.com/fwlink/?LinkId=91243>.

Configuring Tracking for BizTalk Applications

The BizTalk Server Administration console lets you configure various tracking options for items such as send ports and receive ports. The tracking configuration settings enable you to track inbound and outbound event data, message properties, message bodies, and orchestrations. For more information about configuring tracking for BizTalk applications, see "Configuring Tracking Using the BizTalk Server Administration Console" at <http://go.microsoft.com/fwlink/?LinkId=106851>.

You can also use Health and Activity Tracking (HAT) to view historical or tracked data. For more information, see "Viewing Historical or Tracked Data Using HAT" at <http://go.microsoft.com/fwlink/?LinkId=106852>.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Installation Issues

This topic was last updated on: June 05, 2009

◆ Important

The SQL adapter is available as part of the BizTalk Adapter Pack as well as a separate adapter. If you are accessing this topic to know about installation issues with the SQL adapter that is separate from the BizTalk Adapter Pack, all references to the BizTalk Adapter Pack Setup must be interpreted as SQL adapter Setup.

Installation of the Microsoft BizTalk Adapter Pack copies the product binaries on the computer and registers the bindings for each adapter. This section discusses using troubleshooting techniques to resolve installation errors.

Logging Messages for Setup Actions

The BizTalk Adapter Pack setup program performs the standard task of installing the BizTalk Adapter Pack. Additionally, the setup also performs certain custom actions such as registering the adapter bindings. You can log messages for both the standard as well as custom actions that the setup performs.

- The BizTalk Adapter Pack setup installs the adapter-specific files using an MSI. Therefore, the logging for the setup is the standard MSI logging.
- All logs for the custom actions that the setup program performs are available at %TEMP%\adaptersetup.log. If the tracing to the log file fails, the traces are also available in the event log.

Known Issues

The following are the most common errors you might encounter when installing the BizTalk Adapter Pack, along with their probable cause and resolution.

- [Setup fails to register adapter bindings](#)
- [Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation](#)
- [Invalid binding error while configuring SQL adapter ports in BizTalk Server Administration Console on a 64-bit installation](#)

Setup fails to register adapter bindings

Problem

The Microsoft BizTalk Adapter Pack Setup Wizard fails to register the adapter bindings, but proceeds with the adapter installation.

Cause

This might result due to problems with Windows Communication Foundation (WCF) installation, Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK installation, or the machine.config file being corrupt. The adapter bindings are written to the machine.config file.

Resolution

You should manually register the SQL adapter binding.

To register the adapter binding

1. Navigate to the machine.config file on the computer. For example, on a 32-bit platform, the machine.config is available under <system drive>:\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG.

In this path, <version> is the version of the .NET Framework. For example, for Microsoft .NET Framework 3.5 SP1, the version will be v2.0.50727.

2. Open the file by using a text editor.

3. To register the SQL adapter binding:

a. Search for the element "system.serviceModel" and add the following under it:

```
<client>
  <endpoint binding="sqlBinding" contract="IMetadataExchange" name="mssql" />
</client>
```

b. Search for the element "bindingElementExtensions" under system.serviceModel\extensions.

c. Look for the missing SQL adapter binding. Add the following section under the "bindingElementExtensions" node.

For SQL adapter, add:

```
<add name="sqlAdapter" type="Microsoft.Adapters.Sql.SqlAdapterBindingElementExtensionElement,Microsoft.Adapters.Sql, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

d. Search for the element "bindingExtensions" under system.serviceModel\extensions.

e. Look for the missing SQL adapter binding. Add the following section under the "bindingExtensions" node.

For SQL adapter, add:

```
<add name="sqlBinding" type="Microsoft.Adapters.Sql.SqlAdapterBindingCollectionElement,Microsoft.Adapters.Sql, Version=<version>, Culture=neutral, PublicKeyToken=<public key>" />
```

Note
For information about how to determine the public key and the version, see [Determining the Public Key and Version](#).

4. Save and close the machine.config file.

Determining the Public Key and Version

Perform the following steps to determine the public key for SQL adapter.

To determine the public key

1. Navigate to the Windows directory, typically C:\WINDOWS\assembly.
2. Right-click the DLL for which you want the public key and the version, and then select **Properties**. The following table lists the name of the DLL for SQL adapter.

Adapter	Name of the DLL
SQL adapter	Microsoft.Adapters.Sql

3. On the **General** tab, the value against the **Public Key Token** label specifies the public key for the DLL. Similarly, value against the **Version** label specifies the version number for the DLL.
4. Copy the public key, and then click **Cancel**.

Error while using the Consume Adapter Service add-in or Add Adapter Service Reference plug-in on a 64-bit installation

Problem

Using the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in from Visual Studio on a 64-bit computer running 64-bit version of the BizTalk Adapter Pack results in the following error:

No valid adapters are installed on this machine

Cause

The SQL adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, Visual Studio runs as a 32-bit process and hence when you launch the Consume Adapter Service Add-in from Visual Studio, the plug-in checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

◆ Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

Invalid binding error while configuring SQL adapter ports in BizTalk Server Administration Console on a 64-bit installation

Problem

When you try to configure a port for the adapter in BizTalk Server Administration console, you get the following error:

```
"Unable to create binding configuration element for editing. Check the values of the BindingType and BindingConfiguration properties. (Microsoft.Biztalk.Adapter.Wcf.Converters.CreateBindingException) Unable to get binding type for binding extension "sqlBinding". Verify the binding extension is registered in machine.config."
```

Cause

The SQL adapter is a WCF custom binding, which is registered under System.ServiceModel in the machine.config file. A 64-bit platform has two machine.config files, one used by the 32-bit applications and the other used by the 64-bit applications. So, when you install the 64-bit version of the BizTalk Adapter Pack, the setup wizard registers the bindings in the 64-bit version of the machine.config file. However, BizTalk Server Administration console runs as a 32-bit process and hence when you configure a port for the adapter, it checks for the bindings in the 32-bit version of the machine.config file and fails giving an error.

Resolution

Install both the 32-bit and 64-bit versions of the BizTalk Adapter Pack on a 64-bit WCF LOB Adapter SDK installation.

◆ Important

You must only have a 64-bit WCF LOB Adapter SDK installation. Side-by-side installation of 32-bit and 64-bit WCF LOB Adapter SDK on a single computer is not supported.

See Also

Other Resources

[Troubleshooting](#)

Troubleshooting Operational Issues

This topic was last updated on: June 05, 2009

This section discusses using troubleshooting techniques to resolve operational errors that you might encounter when using Microsoft BizTalk Adapter for SQL Server.

Enabling Tracing

You must enable tracing between the adapter, WCF LOB Adapter SDK, and SQL Server to gather more information about any issues you encounter while using the SQL adapter. For more information about tracing support in the SQL adapter, see [Diagnostic Tracing and Message Logging](#).

Known Issues

The following are the most common errors you might encounter when using the SQL adapter, along with their probable cause and resolution.

- [Error in loading the adapter bindings](#)
- [The SQL adapter does not display in the list of adapters in BizTalk Server Administration console](#)
- [Error while performing operations on a SQL Server database](#)
- [InvalidOperationException with ErrorCode=5 while performing FILESTREAM operations](#)
- [Polling operation does not return any messages even if valid statements are specified for PollingStatement and PolledDataAvailableStatement](#)
- [The adapter fails to insert, update, or delete large volumes of data in a single operation using BizTalk Server](#)
- [Full schema validation in BizTalk Server fails for response messages containing DataSet](#)
- [Error with RootNode TypeName in BizTalk Projects](#)
- [Adapter fails to generate metadata of strongly-typed stored procedure with temporary tables](#)
- [Invalid binding warning when using the adapter in Visual Studio 2010](#)
- [BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host](#)
- [Adapter client throws an exception on performing an operation after the connectivity is restored between the adapter client and the SQL Server database](#)
- [Memory usage and thread count increases when using the adapter in a transacted inbound operation](#)

Error in loading the adapter bindings

Problem

When you try to start the Add Adapter Service Reference Visual Studio Plug-in or the Consume Adapter Service BizTalk Project Add-in, you get the following error:

```
There was an error loading the binding, <binding name>, from your system configuration.ConfigurationErrorsException: Exception has been thrown by the target of an invocation.
```

Cause

When you try to start the Add Adapter Service Reference Plug-in or the Consume Adapter Service Add-in, WCF loads the adapter bindings for all the installed adapters. In turn, the adapter bindings are dependent on the specific client software for the enterprise application. You might face this issue if you did a Typical or Complete installation of the adapter, which installs all the adapters contained in the BizTalk Adapter Pack. However, the LOB client libraries might be installed for only one enterprise application. As a result, the GUI fails to load the bindings for the other adapters.

Resolution

Make sure you do a custom installation of the adapters to install only the adapter you need.

The SQL adapter does not display in the list of adapters in BizTalk Server Administration console

Problem

Unlike the earlier version of the adapters shipped with BizTalk Server, the SQL adapter shipped with BizTalk Adapter Pack does not show up in the list of adapters in the BizTalk Server Administration console.

Cause

The latest SQL adapter is a WCF custom binding. So, although the BizTalk Server Administration console displays the WCF-Custom adapter, it does not display the WCF custom bindings and hence, does not display the WCF-based SQL adapter.

Resolution

You can explicitly add the SQL adapter to the BizTalk Server Administration console by following the steps mentioned in [Adding the SQL Adapter to BizTalk Server Administration Console](#).

Error while performing operations on a SQL Server database

Problem

The adapter gives the following error when performing any operation on a SQL Server database using BizTalk Server.

- **For BizTalk Server 2010**

```
System.ArgumentNullException: Value cannot be null.
```

Cause

The WCF action for the message is not specified. WCF requires a SOAP action to be specified for every operation, which informs the adapter about the operation to be performed on the LOB application.

Resolution

Specify the SOAP action in the send port or as a message context property in a BizTalk orchestration. For instructions, see [Specifying SOAP Action](#). See [Technical Reference](#) to see a list of actions for each operation.

InvalidOperationException with ErrorCode=5 while performing FILESTREAM operations

Problem

You get the following error while using the SQL adapter to perform FILESTREAM operations.

```
System.InvalidOperationException: OpenSqlFileStream returned error.  
ErrorCode:5
```

Cause

You might have specified database credentials to connect to the SQL Server database. To perform FILESTREAM operations, you must always use Windows Authentication. The error code "5" denotes that access is denied because of incorrect credentials. For more information about the different error codes, see <http://go.microsoft.com/fwlink/?LinkId=126619>.

Resolution

Use Windows Authentication to connect to the SQL Server database. In BizTalk Server Administration console, you can do so by leaving the user name and password fields blank in the WCF-Custom or WCF-SQL port configuration dialog box.

Polling operation does not return any messages even if valid statements are specified for PollingStatement and PolledDataAvailableStatement

Problem

Even if valid values are specified for the PollingStatement and PolledDataAvailableStatement binding properties, the adapter does not receive a polling message from SQL Server.

Cause

Verify whether any other transaction has taken a lock on the table that the adapter is polling.

Resolution

If you want to poll a table that is being updated as part of another transaction, you can consider using "with (nolock)" parameter as part of the query specified for PolledDataAvailableStatement binding property to ensure that data is returned even if a lock is imposed by the other transaction. For information about locking in the SQL Server database, see <http://go.microsoft.com/fwlink/?LinkId=126715>.

The adapter fails to insert, update, or delete large volumes of data in a single operation using BizTalk Server

Problem

The SQL adapter fails to insert, update, or delete large volumes of data in a single operation using BizTalk Server.

Cause

Inserting, updating, or deleting large volumes of data may take time and the SQL adapter or the transaction in which the operation is being performed, may time out.

Resolution

• For BizTalk Server 2010

1. Specify the timeout for the WCF adapter in the machine.config. Navigate to the machine.config file under <system drive>\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG and add the excerpt that resembles the following.

```
<configuration>  
  <system.transactions>  
    <machineSettings maxTimeout="02:00:00" />  
  </system.transactions>  
</configuration>
```

With this setting, the WCF adapter timeout is set to 2 hours.

2. Specify the timeout settings for MSDTC transactions in the machine.config. Navigate to the machine.config file under <system drive>\WINDOWS\Microsoft.NET\Framework\<version>\CONFIG and add the excerpt that resembles the following.

```
<system.transactions>  
  <defaultSettings distributedTransactionManagerName="<computer_name>" timeout="02:00:00"/>  
</system.transactions>
```

With this setting, the MSDTC timeout is set to 2 hours. The default value for MSDTC timeout is 10 minutes.

Important

You must make this change on the computers running the adapter client and SQL Server. In the excerpt, replace <computer_name> with the name of computer running the adapter client and SQL Server.

3. Set the **SendTimeout** binding property for the SQL adapter to a fairly large value. For instructions on how to set the binding properties, see [Specifying Binding Properties](#).

Full schema validation in BizTalk Server fails for response messages containing DataSet

Problem

For operations that return a response message containing a DataSet, for example ExecuteReader, full schema validation fails in BizTalk Server.

Resolution

We recommend you to not do a full schema validation for response messages containing a dataset. Instead, you could do the following:

1. Execute the operation once that returns the response message with the schema.
2. Copy the schema from the response message to a .xsd file and add this file to your BizTalk project.

3. Use an xpath query in your orchestration to extract the data from the response message.

Error with `RootNode TypeName` in BizTalk Projects

Problem

In a BizTalk project in Visual Studio, if the schemas generated from the Consume Adapter Service Add-in contains invalid characters or reserved words for the `RootNode TypeName` property, the following error will occur while compiling the project:

```
Node <node reference> - Specify a valid .NET type name for this root node.  
The current .NET type name of this root node is invalid (it is a reserved BizTalk keyword or is an invalid C# identifier).
```

Resolution

1. Right-click the root node referenced in the error and select **Properties**.
2. For the `RootNode TypeName` property, remove any illegal characters or reserved words, for example, dot (.).

Adapter fails to generate metadata of strongly-typed stored procedure with temporary tables

Problem

The adapter fails to generate metadata for strongly-typed stored procedures that include temporary tables in their definition. The adapter gives the following exception.

```
Microsoft.ServiceModel.Channels.Common.MetadataException:  
Retrieval of Operation Metadata has failed while building WSDL at 'TypedProcedure/<schema>/<stored_procedure_name>' --->  
System.Data.SqlClient.SqlException: Invalid object name '<temp_table_name>'.
```

Resolution

The SQL adapter does not support generating metadata for strongly-typed stored procedures that contain temporary tables in their definition. Instead, you should generate metadata for the same procedure from under the **Procedures** node while using the Add Adapter Service Reference Plug-in or Consume Adapter Service Add-in.

Invalid binding warning when using the adapter in Visual Studio 2010

Problem

When you use the adapter to create an application in Visual Studio 2010 and you open the configuration file (app.config) generated by the adapter, you see a warning similar to the following:

```
The element 'bindings' has invalid child element 'sqlBinding'. List of possible elements expected: 'basicHttpBinding, customBinding, ...
```

Cause

This warning appears because the SQL adapter binding, `sqlBinding`, is not a standard binding shipped with the Windows Communication Foundation (WCF).

Resolution

You can safely ignore this warning.

BizTalk Server throws an exception if you use more than one Notification schema in the same application or use the Notification schema across multiple applications on the same host

Problem

BizTalk Server throws an XLANG exception or an exception stating that the application cannot locate the document specification because multiple schemas matched the message type.

Cause

This happens because of either of the following:

- You have generated more than one Notification schema in a BizTalk Server project, deployed it to a BizTalk Server application, and then ran the application to receive notifications from the SQL Server database. Because the Notification schemas are common, there is a conflict between the schemas that are deployed in the BizTalk Server application.
- In case of multiple projects, you have generated a Notification schema for each of the BizTalk Server projects, deployed each project to a separate BizTalk Server application on the same host, and then ran an application or applications to receive notifications from the SQL Server database. Because the schemas and assemblies are accessible across the applications in BizTalk Server, there is a conflict between the common schemas deployed under various BizTalk Server applications and assemblies.

Resolution

Use a single Notification schema file for a BizTalk Server application. If you need to use the Notification schema in multiple BizTalk Server applications on the same host, create an application containing a single Notification schema, and then use the notification schema from all other applications in BizTalk Server.

Adapter client throws an exception on performing an operation after the connectivity is restored between the adapter client and the SQL Server database

Problem

Adapter client throws the following exception on executing an operation on the SQL Server database:

```
{System.Data.Common.DbException} = {"A transport-level error has occurred when sending the request to the server. (provider: TCP Provider  
, error: 0 - An existing connection was forcibly closed by the remote host.)"}
```

Cause

During the execution of an operation, the adapter uses the connection from the SQL ADO.NET connection pool to connect to the SQL Server database, and perform the operation. If there is a brief network outage between the adapter client and the SQL Server database or if the SQL Server database is down temporarily, all the connections in the SQL ADO.NET connection pool become invalid. After the connectivity is restored and you try to perform an operation on the SQL Server database, the adapter uses the same invalid connections from the SQL ADO.NET connection pool, and therefore the adapter client throws the exception.

Resolution

The adapter client should implement retry logic in their operation execution where they should catch the exception and specify the operation retry count as "n+1", where "n" is the value specified for the `MaxConnectionPoolSize` binding property. This implies that if there are "n" number of connections in the connection pool that have been rendered invalid, theoretically the adapter client should retry for a maximum of "n+1" times to get a valid connection, and hence perform the operation.

For example, to specify the retry count in BizTalk Server, open the **Properties** dialog box of a send port in an application, click **Transport Advanced Options** in the left pane of the dialog box, and in the **Transport Options** area, specify a value in the **Retry count** list.

Memory usage and thread count increases when using the adapter in a transacted inbound operation

Problem

In a transacted inbound operation, such as Polling, **if there is no data available in the table being polled** and the adapter continues to poll, over a period of time you experience an increase in the memory usage and the thread count.

Cause

If there is no data available in the table being polled, after every receive timeout cycle, Windows Communication Foundation (WCF) spawns a new thread to continue the polling operation. Hence, the thread count and memory usage increases over a period of time. However, if the table being polled has some data, the same thread continues to perform all subsequent polls.

Resolution

We recommend setting the **ReceiveTimeout** to the maximum possible value, which is 24.20:31:23.6470000 (24 days) so that a new thread is spawned only every 24 days. This will ensure that the memory usage and thread count does not grow too much too soon.

For more information about the **ReceiveTimeout** binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). For instructions on specifying binding properties, see [Specifying Binding Properties](#).

Note

When using the adapter with BizTalk Server, setting the timeout to a large value does not impact the functionality of the adapter.
--

See Also

Other Resources

[Troubleshooting](#)

Using Performance Counters

BizTalk Adapter Pack clients can use the performance counters to gauge the performance of the adapters. The BizTalk Adapter Pack setup program creates the performance counter category "Microsoft BizTalk Adapter for SQL Server" along with the Adapter Pack installation.

The LOB Time (Cumulative) Performance Counter

The **BizTalk .NET Adapter for SQL** category has one performance counter called "LOB Time (Cumulative)". This performance counter denotes the time, in milliseconds, that the SQL Server client library takes to complete an action that the adapter initiates. The SQL adapter creates an instance of the performance counter for each action, for a specific SQL Server instance and database name. The instances are created in the following pattern:

```
<processId>:<appDomainId>:<endpointId>:<actionId>
```

The <endpointId> is derived as <sql_server_name>, <instance_name>, <database_name>.

The <actionId> is derived in the following manner:

- For opening a connection, the action ID is "Open".
- For inbound operations, the action ID is "Inbound".
- For outbound operations, the action ID is the action of the operation being invoked, with "/" replaced by an underscore "_". Also, the action ID is prefixed with the "ExecuteScalar", "ExecuteReader", or "ExecuteNonQuery" depending on the method that the adapter internally uses to perform the operation on the SQL Server database. For example, the adapter internally uses the **ExecuteReader** method to execute a stored procedure in SQL Server. So, the action ID for the stored procedure, MyProcedure, will be:

```
ExecuteReader_Procedure_dbo_MyProcedure
```

The performance counter is initialized only after the adapter makes the first call to the SQL Server database. Also, the **InstanceLifetime** property of the performance counter is set to 'Process', which means that the performance counter ceases to exist as soon as the program that creates the counter terminates. For more information about the **InstanceLifetime** property, see <http://go.microsoft.com/fwlink/?LinkId=104181>.

Note

The precision of the LOB Time (Cumulative) performance counter is 16 milliseconds.

Enabling Performance Counters

The performance counters can be enabled or disabled by setting the binding property **EnablePerformanceCounters**. To enable performance counters, set the **EnablePerformanceCounters** binding property to **True**. To disable performance counters, set **EnablePerformanceCounters** to **False**. By default, the property is set to **False**. For more information about this binding property, see [Working with BizTalk Adapter for SQL Server Binding Properties](#).

Performance Counters and the WCF LOB Adapter SDK

Changing the value of the **EnablePerformanceCounters** binding property also changes the value of the corresponding performance counter for the WCF LOB Adapter SDK. Also, the binding property for the WCF LOB Adapter SDK is static, whereas that for the SQL adapter is dynamic. Hence, if there are two instances of the SQL adapter binding in the application domain, and the **EnablePerformanceCounters** binding property is set to **True** in one and **False** in the other, the adapter-specific performance counter will be enabled in one and disabled in the other. However, because the binding property for WCF LOB Adapter SDK is static, it will either be set to **True** or **False** depending on what value was specified last.

See Also

Other Resources

[Troubleshooting](#)

Technical Reference

This section covers technical reference information about Microsoft BizTalk Adapter for SQL Server.

In This Section

- [Messages and Message Schemas for BizTalk Adapter for SQL Server](#)
- [Metadata Node IDs](#)
- [Glossary](#)

Messages and Message Schemas for BizTalk Adapter for SQL Server

The Microsoft BizTalk Adapter for SQL Server is a Windows Communication Foundation (WCF) custom binding. It exposes operations that applications can invoke on it and that it can, in turn, invoke on applications. These operations are invoked by sending SOAP messages over a channel. If a response is required, it is returned in a SOAP message over the same channel.

As a WCF service, the SQL adapter exposes metadata for its operations and data types by using standard WCF mechanisms. The sections in this topic describe the XML structure of the messages and data types that the SQL adapter uses.

In This Section

- [Basic SQL Server Data Types](#)
- [Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views](#)
- [Message Schemas for Procedures and Functions](#)
- [Message Schemas for the Polling and TypedPolling Operations](#)
- [Message Schemas for Query Notification](#)
- [Message Schemas for Composite Operations](#)
- [Message Schemas for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#)

See Also

Other Resources

[Technical Reference](#)

Basic SQL Server Data Types

This topic describes how the Microsoft BizTalk Adapter for SQL Server surfaces basic SQL Server data types.

Supported SQL Server Data Types

The following table shows how the SQL Server data types are surfaced by the SQL adapter:

SQL Server Data Type	XSD type	.NET type	Comments
Bigint	Long	Long	-
Binary	Base64Binary	Byte[]	-
Bit	Boolean	Bool	-
Char	String	String	-
Date	DateTime	DateTime	-
Datetime	DateTime	DateTime	<p>While writing data to a Datetime field, the adapter always stores the time in GMT. If you specify the time-zone information, the adapter uses that to convert the value to a valid GMT value, and writes it to the database table. For example, 12/31/2008T23:59:59+5:30 is written to the table as 12/31/2008 6:29:59 PM.</p> <p>However, if you do not specify the time-zone information, the adapter considers the value to be in GMT already, and writes the same value to the table. For example, 12/31/2008T23:59:59 is written to the table as 12/31/2008 11:59:59 PM.</p>
Datetime2	DateTime	DateTime	-
Datetimeoffset	DateTime	DateTime	-
Decimal	xsd:decimal if precision <= 28 xsd:string if precision > 28	Decimal if precision <= 28 String if precision > 28	-
FileStream	Base64Binary	Byte[]	-
Float	Double	Double	-
Geography	String	String	-

Geometry	String	String	-
Hierarchyid	String	String	-
Image	Base64Binary	Byte[]	-
Int	Int	Int	-
Money	Decimal	Decimal	-
Nchar	String	String	-
Ntext	String	String	-
Numeric	Decimal	Decimal	-
Nvarchar	String	String	-
Nvarchar(Max)	String	String	-
Real	Float	Float	-
Smalldatetime	DateTime	DateTime	-
Smallint	Short	Short	-
Smallmoney	Decimal	Decimal	-
SQLVariant	String	String	-
Text	String	String	-
Time	Duration	TimeSpan	-
Timestamp	Base64Binary	Byte[]	-
Tinyint	UnsignedByte	Byte	-

Unique Identifier	{http://schemas.microsoft.com/2003/10/Serialization/};guid	Guid	-
Varbinary	Base64Binary	Byte[]	-
Varbinary(Max)	Base64Binary	Byte[]	-
Varchar	String	String	-
Varchar(Max)	String	String	-
XML	String	String	-

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for SQL Server](#)

Message Schemas for Insert, Update, Delete, and Select Operations on Tables and Views

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for SQL Server surfaces Insert, Update, Delete, and Select operations for each table and view in the SQL Server database. These operations perform the appropriate SQL statement qualified by a WHERE clause. The SQL adapter uses strongly-typed records and record sets in these operations.

Message Structure for Table Operations

The following table shows the XML message structure for the basic table operations exposed by the SQL adapter on SQL Server database tables. The target table for an operation is specified in the message action and also appears in the target namespace.

Operation	XML Message	Description	SQL Executed by the Adapter
Insert	<pre><Insert xmlns=" [VERSION]/TableOp/[SCHEMA]/[TABLE_NAME]"> <Rows> <[TABLE_NAME]> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </[TABLE_NAME]> </Rows> </Insert></pre>	<p>Inserts the supplied record set of strongly-typed data into the target table.</p>	<pre>INSERT INTO TABLE_NAME (FIELD1_NAME, FIELD2_NAME, ...) VALUES (value1, value2, ...);</pre>
Insert Response	<pre><InsertResponse xmlns=" [VERSION]/TableOp/[SCHEMA]/[TABLE_NAME]"> <InsertResult> <long>[Value]</long> </InsertResult> </InsertResponse></pre>	<p>The Insert Response message contains an array of Long data type. The array stores the identity values of the inserted rows, if any. If there is no identity column in a table, the return value is NULL.</p>	

Select	<p>Selecting all records:</p> <pre><Select xmlns=" [VERSION]/Table Op/[SCHEMA]/[TA BLE_NAME]"> <Columns>*</C OLUMNS> <Query></Quer y> </Select></pre>	<p>A SELECT query is performed on the target table using the WHERE clause specified in the element. The result set contains the columns in the comma-separated list of column names specified in the <Columns> element.</p> <p>It is mandatory to provide value in the <Columns> element. If all columns have to be retrieved in a table or view, * must be specified in the <Columns> element. If specific columns have to be retrieved, the column names must be separated by comma, and specified in the same order as they are defined in the table or view.</p>	<p>Selecting all records:</p> <pre>SELECT * FROM [TABL E_NAME] WHERE [WHER E_clause];</pre>
	<p>Selecting specific columns in a set of records:</p> <pre><Select xmlns=" [VERSION]/Table Op/[SCHEMA]/[TA BLE_NAME]"> <Columns>[COL UMN_list]</COLU MNS> <Query>where [WHERE_clause]< /Query> </Select></pre>	<p>It is mandatory to include the WHERE clause in the SELECT statement. If you do not want to specify a WHERE clause, you can either delete the <Query> element or leave it empty.</p> <p>You can update records using the Select operation. An UPDATE statement is placed in the <Query> element of the SELECT request XML, separated from the WHERE clause by a semicolon. Note that the UPDATE statement does not operate on the result set of the SELECT statement.</p>	<p>Selecting specific columns in a set of records:</p> <pre>SELECT [COLUMN_list] FROM [TABLE_NAME] WHERE [WHERE_clause];</pre> <p>Updating records as part of the Select Operation:</p> <pre>SELECT [COLUMN_list] FROM [TABLE_NAME] WHERE [WHERE_clause]; UPDATE [TABLE_NA ME] SET [FIELD1_NAM E] = value1 [WHERE_ clause];</pre>
	<p>Updating records as part of the Select Operation:</p> <pre><Select xmlns=" [VERSION]/Table Op/[SCHEMA]/[TA BLE_NAME]"> <Columns>[COL UMN_list]</Colu mns> <Query>where [WHERE_clause]; UPDATE [TABLE_N AME] SET [FIELD 1_NAME] = [valu e1] where [WHER E_clause]</Quer y> </Select></pre>		

Select Response	<pre> <SelectResponse xmlns="[VERSION] /TableOp/[SCHEMA]/[TABLE_NAME]"> <SelectResult> <[TABLE_NAME]> <[FIELD1_NAME]>[value1]< /[FIELD1_NAME]> <[FIELD2_NAME]>[value2]< /[FIELD2_NAME]> ... </[TABLE_NAME]> </SelectResult> </SelectResponse> </pre>	The strongly-typed result set generated by the SELECT query --	
Update	<pre> <Update xmlns="[VERSION]/TableOp/[SCHEMA]/[TABLE_NAME]"> <Rows> <RowPair> <After> <[FIELD1_NAME]>[value1]< /[FIELD1_NAME]> <[FIELD2_NAME]>[value2]< /[FIELD2_NAME]> ... </After> <Before> <[FIELD1_NAME]>[value3]< /[FIELD1_NAME]> <[FIELD2_NAME]>[value4]< /[FIELD2_NAME]> ... </Before> </RowPair> </Rows> </Update> </pre>	<p>Take an array of record pairs as input. Each record pair is a collection of two strongly-typed records:</p> <ul style="list-style-type: none"> • First record (in the <After> element) corresponds to new values that need to be updated. • Second record (in the <Before>) corresponds to the old values of the rows. 	<pre> UPDATE [TABLE_NAME] SET [FIELD1_NAME] = value1, [FIELD2_NAME] = value2, ... WHERE [FIELD1_NAME] = v alue3, [FIELD2_NAME] = value4, ...; </pre>

Update Response	<pre><UpdateResponse xmlns="[VERSION]"/TableOp/[SCHEMA]/[TABLE_NAME]"> <UpdateResult>[rows updated] </UpdateResult> </UpdateResponse></pre>	The number of rows updated is returned in the UpdateResult element.	--
Delete	<pre><Delete xmlns="[VERSION]"/TableOp/[SCHEMA]/[TABLE_NAME]"> <Rows> <[TABLE_NAME]> <[FIELD1_NAME]>value1</[FIELD1_NAME]> <[FIELD2_NAME]>value2</[FIELD2_NAME]> ... </[TABLE_NAME]> </Rows> </Delete></pre>	--	<pre>DELETE FROM [TABLE_NAME] WHERE [FIELD1_NAME] = value1, [FIELD2_NAME] = value 2, ...;</pre>
Delete Response	<pre><DeleteResponse xmlns="[VERSION]"/TableOp/[SCHEMA]/[TABLE_NAME]"> <DeleteResult>[rows deleted] </DeleteResult> </DeleteResponse></pre>	The number of rows deleted is returned in the DeleteResult element.	--

[VERSION] = The message version string; for example, <http://schemas.microsoft.com/Sql/2008/05>.

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[TABLE_NAME] = Name of the table; for example, Employee.

[FIELD1_NAME] = Table field name; for example, NAME.

[COLUMN_list] = Comma-separated list of columns; for example, Name, Age, Designation.

[SELECT_query] = A SQL SELECT statement specified in the QUERY element of a Bulk Insert operation; for example, "SELECT * from MyTable"

[WHERE_clause] = WHERE_clause for the SELECT statement used for the operation; for example, ID > 10.

◆ Important

The message structure for the basic table operations on views is the same as that on tables except that the view replaces the table: `Insert xmlns="[VERSION]"/ViewOp/[SCHEMA]/[VIEW_NAME]"`.

Message Actions for Basic Table Operations

The following table shows the message actions that are used by the SQL adapter for the basic table operations on tables. The

SQL adapter uses the table name specified in the message action to determine the target table of the operation.

Operation	Message Action	Example
Insert	TableOp/Insert/[SCHEMA]/[TABLE_NAME]	TableOp/Insert/dbo/Employee
Insert Response	TableOp/Insert/[SCHEMA]/[TABLE_NAME]/response	TableOp/Insert/dbo/Employee/response
Select	TableOp/Select/[SCHEMA]/[TABLE_NAME]	TableOp/Select/dbo/Employee
Select Response	TableOp/Select/[SCHEMA]/[TABLE_NAME]/response	TableOp/Select/dbo/Employee/response
Update	TableOp/Update/[SCHEMA]/[TABLE_NAME]	TableOp/Update/dbo/Employee
Update Response	TableOp/Update/[SCHEMA]/[TABLE_NAME]/response	TableOp/Update/dbo/Employee/response
Delete	TableOp/Delete/[SCHEMA]/[TABLE_NAME]	TableOp/Delete/dbo/Employee
Delete Response	TableOp/Delete/[SCHEMA]/[TABLE_NAME]/response	TableOp/Delete/dbo/Employee/response

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[TABLE_NAME] = Name of the table; for example, Employee.

◆ Important

The message action for an operation on a view is the same as that for a table except that "ViewOp" replaces "TableOp"; for example, **ViewOp**/Insert/dbo/Employee_View.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for SQL Server](#)

Message Schemas for Procedures and Functions

This topic was last updated on: June 05, 2009

The Microsoft BizTalk Adapter for SQL Server surfaces SQL Server database stored procedures and scalar and table valued functions as operations. This section describes the message structure and actions used to invoke procedures and functions.

Message Structure of Procedures and Functions

The operations surfaced for procedures and functions follow a request-response message exchange pattern. The following table shows the structure of these request and response messages.

Operation	XML Message	Description
Stored Procedure Request	<pre><[SP_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/[SCHEMA]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[SP_NAME]></pre>	-
Stored Procedure Response	<pre><[SP_NAME]Response xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/[SCHEMA]"> <[SP_NAME]Result> <DataSet> <any>[Value]</any> <any>[Value]</any> ... </DataSet> </[SP_NAME]Result> <ReturnValue>[Value]</ReturnValue> </[SP_NAME]Response></pre>	The return value of a stored procedure is an array of DataSet.
Strongly-Typed Stored Procedure Request	<pre><[STRNG_SP_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/TypedProcedures/[SCHEMA]"> <[PRM1_NAME]>value1<[/[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[STRNG_SP_NAME]></pre>	-
Strongly-Typed Stored Procedure Response	<pre><[STRNG_SP_NAME]Response xmlns="http://schemas.microsoft.com/Sql/2008/05/TypedProcedures/[SCHEMA]"> <StoredProcedureResultSet0> <StoredProcedureResultSet0 xmlns:ns1="http://schemas.microsoft.com/Sql/2008/05/ProcedureResultSets/[SCHEMA]/[STRNG_SP_NAME]"> <[PRM1_NAME]>value1<[/[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </StoredProcedureResultSet0> </StoredProcedureResultSet0> <ReturnValue>[Value]</ReturnValue> </[STRNG_SP_NAME]Response></pre>	The return value of a strongly-typed stored procedure is an array of strongly-typed data.
Scalar Function Request	<pre><[SCLR_FN_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/ScalarFunctions/[SCHEMA]"> <[PRM_NAME]>value</[PRM_NAME]> </[SCLR_FN_NAME]></pre>	-

Scalar Function Response	<pre><[SCLR_FN_NAME]Response xmlns="http://schemas.microsoft.com/Sql/2008/05/ScalarFunctions/[SCHEMA]"> <[SCLR_FN_NAME]Result>return_value</[SCLR_FN_NAME]Result> <[PRM_NAME]>value</[PRM_NAME]> </[SCLR_FN_NAME]Response></pre>	-
Table Valued Function Request	<pre><[TBL_FN_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/TableValuedFunctions/[SCHEMA]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[TBL_FN_NAME]></pre>	-
Table Valued Function Response	<pre><[TBL_FN_NAME]Response xmlns="http://schemas.microsoft.com/Sql/2008/05/TableValuedFunctions/[SCHEMA]"> <[TBL_FN_NAME]Result> <[TBL_FN_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/TableValuedFunctions/[SCHEMA]"> <[PRM1_NAME]>value1</[PRM1_NAME]> <[PRM2_NAME]>value2</[PRM2_NAME]> ... </[TBL_FN_NAME]"> ... </[TBL_FN_NAME]Result> </[TBL_FN_NAME]Response></pre>	

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[SP_NAME] = The stored procedure to be executed; for example, ADD_EMP_DETAILS.

[STRNG_SP_NAME] = The strongly-typed stored procedure to be executed; for example, GET_EMP_DETAILS.

[SCLR_FN_NAME] = The scalar function to be executed; for example, GET_EMP_ID.

[TBL_FN_NAME] = The table valued function to be executed; for example, TVF_EMPLOYEE.

[PRM_NAME] = The name of the SQL Server parameter.

Message Actions of Functions and Procedures

The SQL adapter uses the following message actions for stored procedure and function operations.

Message	Action	Example
Stored Procedure Request	Procedure/[SCHEMA]/[SP_NAME]	Procedure/dbo/ADD_EMP_DETAILS
Stored Procedure Response	Procedure/[SCHEMA]/[SP_NAME]/response	Procedure/dbo/ADD_EMP_DETAILS/response
Strongly-Typed Stored Procedure Request	TypedProcedure/[SCHEMA]/[STRNG_SP_NAME]	TypedProcedure/dbo/GET_EMP_DETAILS
Strongly-Typed Stored Procedure Response	TypedProcedure/[SCHEMA]/[STRNG_SP_NAME]/response	TypedProcedure/dbo/GET_EMP_DETAILS/response
FOR XML Stored Procedure Request	XmlProcedure/[SCHEMA]/[SP_NAME]	XmlProcedure/dbo/GET_EMP_DETAILS_FOR_XML

FOR XML Stored Procedure Response	XmlProcedure/[SCHEMA]/[SP_NAME]/resp	XmlProcedure/dbo/GET_EMP_DETAILS_FOR_XML/response
Scalar Function Request	ScalarFunction/[SCHEMA]/[SCLR_FN_NAME]	ScalarFunction/dbo/GET_EMP_ID
Scalar Function Response	ScalarFunction/[SCHEMA]/[SCLR_FN_NAME]/response	ScalarFunction/dbo/GET_EMP_ID/response
Table Valued Function Request	TableFunction/[SCHEMA]/[TBL_FN_NAME]	TableFunction/dbo/TVF_EMPLOYEE
Table Valued Function Response	TableFunction/[SCHEMA]/[TBL_FN_NAME]/response	TableFunction/dbo/TVF_EMPLOYEE/response

[SP_NAME] = The stored procedure to be executed; for example, ADD_EMP_DETAILS.

[STRNG_SP_NAME] = The strongly-typed stored procedure to be executed; for example, GET_EMP_DETAILS.

[SCLR_FN_NAME] = The scalar function to be executed; for example, GET_EMP_ID.

[TBL_FN_NAME] = The name of the table valued function to be executed; for example, TVF_EMPLOYEE.

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for SQL Server](#)

Message Schemas for the Polling and TypedPolling Operations

The Microsoft BizTalk Adapter for SQL Server surfaces the Polling and TypedPolling inbound operations to return the result set of the polling query to an adapter client.

You configure the Polling and TypedPolling operations by setting binding properties in the SQL adapter. For more information about these binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). You set the **PollingStatement** binding property to specify a SQL statement (SELECT or EXEC <stored procedure>) for the polling query. The result set of this query is returned as data to your code in the Polling operation, and as strongly-typed data in the TypedPolling operation. The structure of the result set is determined by the metadata that the adapter surfaces for the specified query.

Message Structure for the Polling and TypedPolling Operations

The following table shows the XML message structure for the Polling and TypedPolling operations.

Operation	XML Message	Description
Polling	<pre><?xml version="1.0" encoding="utf-8" ?> <Polling xmlns="http://schemas.microsoft.com/Sql/2008/05/Polling"> <PolledData> <DataSet xmlns="http://schemas.datacontract.org/2004/07/System.Data"> <any>[Value]</any> <any>[Value]</any> ... </DataSet> </PolledData> </Polling></pre>	This is the inbound message that is sent by the SQL Server to the adapter clients.
Typed Polling	<pre><?xml version="1.0" encoding="utf-8" ?> <TypedPollingResultSet xmlns="http://schemas.microsoft.com/Sql/2008/05/TypedPolling"> <COLUMN1>[Value]</Column1> <COLUMN2>[Value]</Column2> ... </TypedPollingResultSet></pre>	This is the strongly-typed inbound message that is sent by the SQL Server to the adapter clients.

Message Action for the Polling and TypedPolling Operations

The message action for the:

- Polling operation is "Polling."
- TypedPolling operation is "TypedPolling."

See Also

Other Resources

[Technical Reference](#)

Message Schemas for Query Notification

The Microsoft BizTalk Adapter for SQL Server surfaces the Notification operation to receive query notifications from the SQL Server database.

You configure the Notification operation by setting binding properties in the SQL adapter. For more information about the Notification-related binding properties, see [Working with BizTalk Adapter for SQL Server Binding Properties](#). You set the **NotificationStatement** binding property to specify a SQL statement (SELECT or EXEC <stored procedure>) for the query notification. The result set of this query is returned as strongly-typed data to your code in the Notification operation.

Message Structure for the Notification Operation

The following table shows the XML message structure for the Notification operation.

Operation	XML Message	Description
Notification	<pre data-bbox="172 725 451 1178"><?xml version="1.0" encoding="utf-8"?> <Notification xmlns="http://schemas.microsoft.com/Sql/2008/05/Notification"> <Info>Value</Info> <Source>Value</Source> <Type>Value</Type> </Notification></pre>	<p data-bbox="496 725 1497 786">This is the inbound message that is sent by the SQL Server to the adapter clients. In the message:</p> <ul data-bbox="528 819 1501 1256" style="list-style-type: none"> <li data-bbox="528 819 1501 920">• The <Info> tag indicates the reason for the notification. For example, an "insert" value in this tag indicates that data has been inserted in one or more of the tables referenced in the notification statement. <li data-bbox="528 965 1501 1066">• The <Source> tag indicates the source for the notification. For example, a "data" value in this tag indicates a change in the data in a referenced object. Similarly, an "object" value in this tag indicates a change in a referenced object. <li data-bbox="528 1111 1501 1256">• The <Type> tag indicates the type of data change. Query notification messages are of two types: change and subscribe. A "change" value in the <Type> tag indicates that the results of the query have changed, whereas a "subscribe" value in the <Type> tag indicates that a subscription request failed. <p data-bbox="496 1323 1501 1384">For more information about the query notification message tags, see the "Properties" section at http://go.microsoft.com/fwlink/?LinkId=122393.</p>

Message Action for the Notification Operation

The message action for the notification operation is "Notification."

See Also

Other Resources

[Technical Reference](#)

Message Schemas for Composite Operations

The Microsoft BizTalk Adapter for SQL Server enables you to execute composite operations on the SQL Server database. A composite operation can contain multiple operations including the Insert, Update, and Delete operations on the tables and views, and operations on stored procedures. A composite operation can include these operations in any order.

For more information about:

- Composite operations, see [Support for Composite Operations](#).
- How to perform composite operations using the SQL adapter, see [Performing Composite Operations on SQL Server by Using BizTalk Server](#).

Message Structure for the Composite Operation

Since a composite operation contains multiple individual operations; the message structure of a composite operation contains message structures of the individual operations. As a composite operation contains operations on tables, views, and stored procedures, the composite operation message follows a request-response message exchange pattern.

The following table shows the structure of the request and response messages of a composite operation that contains an Insert operation, a stored procedure that does not take any input parameters, and a Delete operation.

Operation	XML Message
Composite Operation Request	<pre> <?xml version="1.0" encoding="utf-8" ?> <Request xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <Insert xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/[SCHEMA A]/[TABLE_NAME]"> <Rows> <[TABLE_NAME]> <[FIELD1_NAME]>[Value1]</[FIELD1_NAME]> <[FIELD2_NAME]>[Value1]</[FIELD2_NAME]> ... </[TABLE_NAME]> </Rows> </Insert> <[SP_NAME] xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/ [SCHEMA]" /> <Delete xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/[SCHEM A]/[TABLE_NAME]"> <Rows> <[TABLE_NAME]> <[FIELD1_NAME]>[Value1]</[FIELD1_NAME]> </[TABLE_NAME]> </Rows> </Delete> </Request> </pre>

Composite Operation Response	<pre> <?xml version="1.0" encoding="utf-8" ?> <RequestResponse xmlns="http://[PROJECT_NAME].[COMPOSITE_SCHEMA_NAME]"> <InsertResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/[SCHEMA]/[TABLE_NAME]"> <InsertResult> <long>[value]</long> </InsertResult> </InsertResponse> <[SP_NAME]Response xmlns="http://schemas.microsoft.com/Sql/2008/05/Procedures/[SCHEMA]"> <[SP_NAME]Result> <DataSet> <any>[Value]</any> <any>[Value]</any> ... </DataSet> </[SP_NAME]Result> <ReturnValue>[value]</ReturnValue> </[SP_NAME]Response> <DeleteResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/TableOp/[SCHEMA]/[TABLE_NAME]"> <DeleteResult>[value]</DeleteResult> </DeleteResponse> </RequestResponse> </pre>
------------------------------	---

[PROJECT_NAME] = Name of the BizTalk project that contains the composite operation schema.

[COMPOSITE_SCHEMA_NAME] = Name of the composite operation schema given by the user.

[SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[TABLE_NAME] = Name of the table; for example, Employee.

[FIELD1_NAME] = Table field name; for example, NAME.

[SP_NAME] = The stored procedure to be executed; for example, ADD_EMP_DETAILS.

Message Action for the Composite Operation

The message action for the composite operation is "CompositeOperation."

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for SQL Server](#)

Message Schemas for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The Microsoft BizTalk Adapter for SQL Server exposes the ExecuteNonQuery, ExecuteReader, and ExecuteScalar outbound operations at the root level to execute any arbitrary SQL statements in SQL Server.

For more information about:

- These operations, see [Support for Executing ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations](#).
- Performing these operations using the SQL adapter, see [Performing ExecuteReader, ExecuteScalar, or ExecuteNonQuery Operations by Using BizTalk Server](#).

Message Structure for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The messages in these operations follow a request-response message exchange pattern, and the following table shows the structure of these request and response messages.

Operation	XML Message	Description
Execute NonQuery Request	<pre><ExecuteNonQuery xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <Query>[PL/SQL STATEMENT1];[PL/SQL STATEMENT2];...</Query> </ExecuteNonQuery></pre>	Within the <Query> tag, you can specify multiple PL/SQL statements separated by a semi-colon.
Execute NonQuery Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteNonQueryResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <ExecuteNonQueryResult>[value]</ExecuteNonQueryResult> </ExecuteNonQueryResponse></pre>	For the UPDATE, INSERT, and DELETE statements, [value] represents the number of rows affected by the PL/SQL statements in the <i>ExecuteNonQuery Request</i> message. For all other types of statements, [value] is -1.
Execute Reader Request	<pre><ExecuteReader xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <Query>[PL/SQL STATEMENT1];[PL/SQL STATEMENT2];...</Query> </ExecuteReader></pre>	Within the <Query> tag, you can specify multiple PL/SQL statements separated by a semi-colon.

Execute Reader Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteReaderResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <ExecuteReaderResult> <DataSet> <Any>[value]</Any> > <Any>[value]</Any> > ... </DataSet> </ExecuteReaderResult> </ExecuteReaderResponse> </></pre>	The result set is the response message of the PL/SQL statements executed in the <i>ExecuteReader Request</i> message, and is returned as an array of DataSet. For information about DataSet, see "DataSet Class" at http://go.microsoft.com/fwlink/?LinkID=196853 .
Execute Scalar Request	<pre><ExecuteScalar xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <Query>[PL/SQL STATEMENT1];[PL/SQL STATEMENT2];...</Query> </ExecuteScalar></pre>	Within the <code><Query></code> tag, you can specify multiple PL/SQL statements separated by a semi-colon.
Execute Scalar Response	<pre><?xml version="1.0" encoding="utf-8" ?> <ExecuteScalarResponse xmlns="http://schemas.microsoft.com/Sql/2008/05/GenericTableOp/"> <ExecuteScalarResult> [value]</ExecuteScalarResult> </ExecuteScalarResponse> </></pre>	The <code>[value]</code> represents the value in the first column of the first row in the result set returned by the PL/SQL statements in the <i>ExecuteScalar Request</i> message.

[PL/SQL STATEMENT] = The entire PL/SQL statement to be executed.

Message Action for the ExecuteNonQuery, ExecuteReader, and ExecuteScalar Operations

The following table shows the message actions that are used by the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operations.

Operation	Action
ExecuteNonQuery Request	GenericOp/ExecuteNonQuery
ExecuteNonQuery Response	GenericOp/ExecuteNonQuery/response
ExecuteReader Request	GenericOp/ExecuteReader
ExecuteReader Response	GenericOp/ExecuteReader/response
ExecuteScalar Request	GenericOp/ExecuteScalar

ExecuteScalar Response	GenericOp/ExecuteScalar/response
------------------------	----------------------------------

See Also

Other Resources

[Messages and Message Schemas for BizTalk Adapter for SQL Server](#)

Metadata Node IDs

The Microsoft BizTalk Adapter for SQL Server surfaces SQL Server database artifacts in a hierarchical manner. The following table lists the node types and node IDs for SQL Server database artifacts that the SQL adapter surfaces. The node ID is the absolute path of the node that is used in the **IMetadataRetrievalContract Browse**, **Search**, and **GetMetadata** methods.

Artifact Display Name	Node Type	Node ID	Example	Description
--	CATEGORY	/	/	WCF LOB Adapter SDK root node. Returns all first-level nodes; this includes the ExecuteNonQuery, ExecuteReader, and ExecuteScalar operation nodes and all schema nodes for the outbound operations, and the Polling operation node for the inbound operation.
ExecuteNonQuery	OUTBOUND OPERATION	GenericOp/ExecuteNonQuery	GenericOp/ExecuteNonQuery	ExecuteNonQuery operation node. Returns WSDL for the ExecuteNonQuery operation.
ExecuteReader	OUTBOUND OPERATION	GenericOp/ExecuteReader	GenericOp/ExecuteReader	ExecuteReader operation node. Returns WSDL for the ExecuteReader operation.
ExecuteScalar	OUTBOUND OPERATION	GenericOp/ExecuteScalar	GenericOp/ExecuteScalar	ExecuteScalar operation node. Returns WSDL for the ExecuteScalar operation.
Polling	INBOUND OPERATION	Polling	Polling	Polling operation node. Returns WSDL for the Polling operation.
Notification	INBOUND OPERATION	Notification	Notification	Notification operation node. Returns WSDL for the Notification operation.
Procedures	CATEGORY	Procedures/	Procedures/	Schema procedures node. Returns all procedures for the specified schema.
[DB_PROCEDURE]	OUTBOUND OPERATION	Procedure/[DB_SCHEMA]/[Procedure_Name]	Procedure/db o/ADD_EMP_DETAILS	Procedure node. Returns the WSDL for the specified procedure.
Tables	CATEGORY	Tables/	Tables/	Schema tables node. Returns all table nodes for the specified schema.
[DB_TABLE]	CATEGORY	-	-	Table node. Returns all operation nodes (Insert, Select, Update, Delete, and Set) for the specified table. The Set operation is only returned for tables that contain columns with any of the following data type: Varchar(Max), Nvarchar(Max) or Varbinary(Max).

Insert	OUTBOUND OPERATION	TableOp/Insert/[DB_SCHEMA]/[DB_TABLE]	TableOp/Insert/dbo/Employee	Table Insert operation node. Returns WSDL for the Insert operation for the specified table.
Select	OUTBOUND OPERATION	TableOp/Select/[DB_SCHEMA]/[DB_TABLE]	TableOp/Select/dbo/Employee	Table Select operation node. Returns WSDL for the Select operation for the specified table.
Update	OUTBOUND OPERATION	TableOp/Update/[DB_SCHEMA]/[DB_TABLE]	TableOp/Update/dbo/Employee	Table Update operation node. Returns WSDL for the Update operation for the specified table.
Delete	OUTBOUND OPERATION	TableOp/Delete/[DB_SCHEMA]/[DB_TABLE]	TableOp/Delete/dbo/Employee	Table Delete operation node. Returns WSDL for the Delete operation for the specified table.
Set[COLUMN_NAME]	OUTBOUND OPERATION	TableOp/WriteText/[DB_SCHEMA]/[DB_TABLE]/[COLUMN_NAME]	TableOp/WriteText/dbo/Employee/Job_Description	Table Set operation node. Returns WSDL for the Set operation for the specified column in the table. (Only surfaced if the table contains columns with any of the following data type: (Max), Nvarchar(Max) or Varbinary(Max)).
Views	CATEGORY	Views/	Views/	Schema views node. Returns all view nodes for the specified schema.
[DB_VIEW]	CATEGORY	-	-	View node. Returns all operation nodes (Insert, Select, Update, and Delete) for the specified view.
Insert	OUTBOUND OPERATION	ViewOp/Insert/[DB_SCHEMA]/[DB_VIEW]	ViewOp/Insert/dbo/Employee_View	View Insert operation node. Returns WSDL for the Insert operation for the specified view.
Select	OUTBOUND OPERATION	ViewOp/Select/[DB_SCHEMA]/[DB_VIEW]	ViewOp/Select/dbo/Employee_View	View Select operation node. Returns WSDL for the Select operation for the specified view.
Update	OUTBOUND OPERATION	ViewOp/Update/[DB_SCHEMA]/[DB_VIEW]	ViewOp/Update/dbo/Employee_View	View Update operation node. Returns WSDL for the Update operation for the specified view.
Delete	OUTBOUND OPERATION	ViewOp/Delete/[DB_SCHEMA]/[DB_VIEW]	ViewOp/Delete/dbo/Employee_View	View Delete operation node. Returns WSDL for the Delete operation for the specified view.
Scalar Functions	CATEGORY	ScalarFunctions/	ScalarFunctions/	Schema scalar functions node. Returns all scalar functions for the specified schema.

[DB_SCLR_FUNCTION]	OUTBOUND OPERATION	ScalarFunction/[DB_SCHEMA]/[DB_SCLR_FUNCTION]	ScalarFunction/dbo/GET_EMP_ID	Scalar function node. Returns the WSDL for the specified scalar function.
Table Valued Functions	CATEGORY	TableFunctions/	TableFunctions/	Schema table valued functions node. Returns all table valued functions for the specified schema.
[DB_TBL_FUNCTION]	OUTBOUND OPERATION	TableFunction/[DB_SCHEMA]/[DB_TBL_FUNCTION]	TableFunction/dbo/TVF_EMPLOYEE	Table valued function node. Returns the WSDL for the specified table valued function.

[DB_SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[DB_TABLE] = The name of an SQL Server table; for example, Employee.

[DB_VIEW] = The name of an SQL Server view; for example, Employee_View.

[DB_PROCEDURE] = The name of an SQL Server stored procedure; for example, ADD_EMP_DETAILS.

[DB_SCLR_FUNCTION] = The name of an SQL Server scalar function; for example, GET_EMP_ID.

[DB_TBL_FUNCTION] = The name of an SQL Server table valued function; for example, TVF_EMPLOYEE.

Metadata Search and Node IDs

Metadata search is a powerful feature that the WCF LOB Adapter SDK surfaces as part of its **MetadataRetrievalContract** interface. The SQL adapter uses this feature to support searching on the following SQL Server artifacts. The metadata search scope is restricted to the level immediately under the node at which the search operation is performed. For example, to search for a scalar function, you must be searching under /Scalar Function/[Schema]. Recursive search is not supported.

Artifact	Node ID	Node Type Returned	Description
/(i.e. Root node)	/	CATEGORY	Return all schema nodes that match the search expression.
[DB_PROCEDURE]	/Procedure/[DB_SCHEMA]	OUTBOUND OPERATION	Return all procedure nodes in the specified schema that match the search expression.
[DB_TABLE]	/Table/[DB_SCHEMA]	CATEGORY	Return all table nodes in the specified schema that match the search expression.
[DB_VIEW]	/View/[DB_SCHEMA]	CATEGORY	Return all view nodes in the specified schema that match the search expression.
[DB_SCLR_FUNCTION]	/ScalarFunction/[DB_SCHEMA]	OUTBOUND OPERATION	Return all scalar function nodes in the specified schema that match the search expression.
[DB_TBL_FUNCTION]	/TableFunction/[DB_SCHEMA]	OUTBOUND OPERATION	Return all table valued function nodes in the specified schema that match the search expression.

[DB_SCHEMA] = Collection of SQL Server artifacts; for example, dbo.

[DB_TABLE] = The name of an SQL Server table; for example, Employee.

[DB_VIEW] = The name of an SQL Server view; for example, Employee_View.

[DB_PROCEDURE] = The name of an SQL Server procedure; for example, ADD_EMP_DETAILS.

[DB_SCLR_FUNCTION] = The name of an SQL Server scalar function; for example, GET_EMP_ID.

[DB_TBL_FUNCTION] = The name of an SQL Server table valued function; for example, TVF_EMPLOYEE.

You can specify search expressions that are compatible with any valid expression that can be used for the SQL Server LIKE operator. For example, to perform a search on the tables contained in a schema, the SQL adapter executes the following SQL:

```
SELECT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME LIKE '[SEARCH_STR]'.
```

The following table lists the special characters that the SQL adapter supports in search expressions.

Special Character	Interpretation
% (percentage)	Matches zero or more characters. For example, "A%" matches "A", "AB", "ABC", and so on.
_ (underscore)	Matches exactly 1 character. For example, "A_" matches "AB", "AC", "AD", and so on.
[]	<ul style="list-style-type: none">Escapes the special meaning of _ and %.Specifies a range or set of characters to be present. For example: <ul style="list-style-type: none">%[%]% matches all names that include a % symbol.[a-f] matches all names that have characters between and including 'a' and 'f'.[abc] matches all names that have characters 'a', 'b', and 'c'.
[^]	Specifies a range or set of characters not to be present. For example: <ul style="list-style-type: none">[^a-f] matches all names that do not have characters between and including 'a' and 'f'.[^abc] matches all names that do not have characters 'a', 'b', and 'c'.

Metadata Retrieval and Node IDs

The following table summarizes the metadata characteristics returned by SQL adapter.

Artifact	Metadata Characteristics
----------	--------------------------

Table or View	<ul style="list-style-type: none"> • Table name. • Table field names. • Table field data types are mapped to simple or complex WSDL types. • Table field length is mapped to facet maxLength. • Table field primary key constraint is mapped to facet minOccurs = 1. • Table field NULL constraint is mapped to facet isNillable = true. • Table operations <ul style="list-style-type: none"> • INSERT • SELECT • UPDATE • DELETE • SET<Column Name>
Procedure or Function	<ul style="list-style-type: none"> • Procedure or function name is mapped to the operation name. • Procedure or function parameter names. • Procedure or function parameter data types are mapped to WSDL types. • Procedure or function parameter direction is mapped to WSDL parameter direction. • Procedure parameter or function parameter data type length is mapped to facet maxLength. • Procedure or function parameter order is mapped to element sequence. • Function return data type is mapped to WSDL type. • Function return data type length is mapped to facet maxLength.

For detailed information about the format of the metadata that the SQL adapter exposes for specific artifacts and operations on the SQL Server database, see [Messages and Message Schemas for BizTalk Adapter for SQL Server](#).

See Also

Other Resources

[Technical Reference](#)

Glossary

The following terms and definitions are used in Microsoft BizTalk Adapter for SQL Server Help.

Microsoft BizTalk Adapter Pack Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

Term	Definition
adapter	A WCF-based component that helps exchange messages between applications (for example, a line-of-business system) and BizTalk Server. The adapter consists of design-time components and run-time components for receive and send operations.
adapter client	An application that interacts with a line-of-business (LOB) system through the adapter.

B

Term	Definition
binding	A process by which software components and layers are linked together. When a network component is installed, the binding relationships and dependencies for the components are established. Binding allows components to communicate with each other. In BizTalk Server, an established mapping between an orchestration adapter-agnostic endpoint (port or role link) and physical adapter-specific endpoints (send/receive ports or party).
BizTalk Server	Connects diverse software. BizTalk Server enables you to create and modify process logic that uses that software. BizTalk Server also enables information workers to monitor running processes, interact with trading partners, and perform other business-oriented tasks.

C

Term	Definition
channel	A concrete implementation of a binding element. The binding represents the configuration, and the channel is the implementation associated with that configuration. Therefore, there is a channel associated with each binding element. Channels stack on top of each other to create the concrete implementation of the binding: the channel stack.
connection URL	A string that identifies a resource in a distributed environment. Adapters use a connection Uniform Resource Identifier (URI) that contains the information necessary to establish a connection with the LOB system.
contract	Specifies the collection and structure of messages required to access the operations offered by the service.

D

Term	Definition
data manipulation language (DML)	The subset of SQL statements that is used to retrieve and manipulate data. DML statements typically start with SELECT, INSERT, UPDATE, or DELETE.
design-time experience	Procedures and operations that a developer performs during design time; for example, using the Consumer Adapter Service BizTalk Project Add-in to retrieve message schemas.

E

Term	Definition
------	------------

endpoint address	A network address that identifies the location of a Windows Communication Foundation (WCF) service endpoint. For an adapter, the endpoint address is expressed as a connection Uniform Resource Identifier (URI) that contains location and connection parameters. The adapter can use these to establish a connection to the underlying line-of-business (LOB) system.
Enterprise Single Sign-on system	An SSO database, a master secret server, and one or more Enterprise Single Sign-On (SSO) servers. These servers do the mapping between the Windows and non-Windows credentials, look up the credentials in the SSO database, and are used for administering the SSO system. The SSO database is also used as a configuration store to hold custom configuration data for adapters.
Extensible Markup Language	A markup language designed to describe data. XML tags are not predefined.

G

Term	Definition
GAC	See global assembly cache.
global assembly cache (GAC)	A machine-wide code cache that stores assemblies specifically installed to be shared by many applications on the computer. Applications deployed in the global assembly cache must have a strong name.

H

Term	Definition
Health and Activity Tracking (HAT)	A user interface that enables specified data to be tracked. This interface can be used to monitor the health of the BizTalk Server farm and to track and view specific events and messages.

I

Term	Definition
inbound operation	An operation that is invoked by a line-of-business (LOB) system on the adapter.

M

Term	Definition
metadata	In WCF, refers to a description of the contract exposed by a service. This is known as the service description and is expressed in a WSDL document. The metadata exposed by an adapter describes the (interface to) the operations that it can perform on the underlying LOB system.
Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK	The specifications for building BizTalk adapters using open standards based on Web services.

O

Term	Definition
one-way	A message exchange pattern (MEP) in which the sender sends a message, but no response is returned by the receiver. In BizTalk Server, MEPs are referred to as communication patterns.
outbound operation	An operation that is invoked by the adapter on the line-of-business system (LOB).
output.cs	The default output file created by the ServiceModel Metadata Utility tool (svcutil.exe).

P

Term	Definition
polling	A technique that device drivers use to find out from multiple devices whether they contain data to transmit. The devices are polled one at a time.

proxy	In WCF, refers to a managed-code object that implements the service contract exposed by a service. The WCF service model is based on the use of such proxies. In the WCF service model, the service contract is expressed as a .NET interface.
-------	--

R

Term	Definition
request-reply	A message exchange pattern (MEP) in which the sender sends a request message and expects a response message from the receiver. In BizTalk Server, MEPs are referred to as communication patterns. Depending on the messaging technology and the direction of the request message (inbound or outbound), this pattern is also called request-reply or solicit-response.
runtime experience	Procedures and operations performed by a developer during run time or when deploying a solution; for example, creating a physical port binding from the BizTalk Server Administration console.

S

Term	Definition
schema	The structure for a message. A schema can contain multiple subschema.
ServiceModel Metadata Utility Tool (svcutil.exe)	A command-line utility that is included with WCF. It is used to create service model proxy code from the service description (metadata) that is exposed by a WCF service such as an adapter. For outbound operations, the tool creates a WCF client class and helper code; for inbound operations, the tool creates a WCF service contract and helper code.
Simple Object Access Protocol	See SOAP.
SOAP	A simple, XML-based protocol for exchanging structured and type information in decentralized, distributed environments. WCF is based on the exchange of SOAP messages between clients and services to invoke operations and return results.
SOAP message	A well-formed XML document. It should use the SOAP envelope and SOAP encoding namespaces and include an optional XML declaration, followed by a SOAP envelope (the root element), which is made up of an optional SOAP header and a SOAP message body.
SQL Server Integration Services (SSIS)	A component that is used to import, export, and transform data from different data sources. Previously called data transformation service (DTS).
SSIS	See SQL Server Integration Services.
SSO	See Enterprise Single Sign-on system.
strongly-typed data	A data set or result set that is bound to an underlying object type. Each row in a strongly-typed XML data set is composed of typed, named elements that correspond to fields of the underlying object type.
svcutil.exe	See ServiceModel Metadata Utility Tool.
synonym	An alternative name for a schema-scoped object; introduced in SQL Server 2005.

U

Term	Definition
Uniform Resource Identifier (URI)	See connection URI

W

Term	Definition
WCF	See Windows Communication Foundation
WCF channel model	A programming model that relies on several interfaces and other types. Channels provide a low-level programming model for sending and receiving messages.
WCF client	A client-application construct that exposes the service operations as methods. You can use the Add Adapter Service Reference Visual Studio Plug-in or the ServiceModel Metadata Utility Tool to generate a WCF client class from the metadata exposed by an adapter.
WCF LOB Adapter SDK	See Microsoft Windows Communication Foundation (WCF) Line of Business (LOB) Adapter SDK
WCF service contract	A managed-code representation of the service contract. It is expressed as an interface in which classes and methods are attributed to define the service, operation, message, and data contracts used to communicate with a service. You can use the ServiceModel Metadata Utility tool or the Add Adapter Service Reference Visual Studio Plug-in to generate a WCF service contract from the metadata exposed by an adapter. You implement the WCF service contract to receive operations from an LOB system.
WCF service model	A WCF programming model in which a service is represented as a managed code object. The operations exposed by the service are represented as methods with strongly-typed data.
weakly-typed data	A data set or result set that is not bound to an underlying object type. Each row in a weakly-typed XML data set is composed of a collection of generic columns in which attributes describe the name and type of each element.
Web services	A unit of application logic providing data and services to other applications. Applications access XML Web services using standard Web protocols and data formats such as HTTP, XML, and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web , and are a cornerstone of the Microsoft .NET programming model.
Web Services Description Language (WSDL)	An XML-based language that describes a service as a set of endpoints that operate on messages. The WSDL document describes the service contract, operation contracts, message contracts, and data contracts that a client must use to interface with the service.
Windows Communication Foundation (WCF)	A Microsoft service-oriented communication infrastructure. The framework inherently provides clients with a service programming model and a channel programming model for finer control of message exchanges.
WSDL	See Web Services Description Language.

X

Term	Definition
XML	See Extensible Markup Language.
XML Schema definition language (XSD)	A schema language. An XML Schema defines the elements, attributes, and data types that comply with the World Wide Web Consortium (W3C) XML Schema Part 1: Structures Recommendation for the XML Schema Definition Language. The W3C XML Schema Part 2: Datatypes Recommendation is the recommendation for defining data types that are used in XML schemas. The XML Schema definition language enables you to define the structure and data types for XML messages.
XSD	See XML Schema definition language.

Class Library

Namespaces

Namespace
Microsoft.Adapters.Sql

Microsoft.Adapters.Sql Namespace

Classes

Class	Description
SqlAdapterBinding	Represents the SQL adapter binding.
SqlAdapterBindingCollectionElement	Describes an element of a SQL adapter binding collection.
SqlAdapterBindingConfigurationElement	Describes a SQL adapter binding configuration element.
SqlAdapterBindingElement	Describes a SQL adapter binding element.
SqlAdapterBindingElementExtensionElement	Represents the SQL adapter binding.
SqlAdapterConnectionUri	Contains the SQL adapter connection URI.
SqlAdapterInboundTransactionBehavior	Describes the SQL adapter inbound transaction behavior.

Enumerations

Enumeration	Description
InboundOperation	Describes the inbound operation.

InboundOperation Enumeration

Describes the inbound operation.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public Enumeration InboundOperation
```

C#

```
public enum InboundOperation
```

C++

```
public enum class InboundOperation
```

J#

```
public enum InboundOperation
```

JScript

```
public enum InboundOperation
```

Members

Member name	Description
Notification	Specifies that the inbound operation type is a notification.
Polling	Specifies that the inbound operation is a polling operation.
TypedPolling	Specifies that the inbound operation is a typed polling operation.
XmlPolling	Specifies that the inbound operation is an XML polling operation.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Class

Represents the SQL adapter binding.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Class SqlAdapterBinding  
    Inherits AdapterBinding
```

C#

```
public class SqlAdapterBinding : AdapterBinding
```

C++

```
public ref class SqlAdapterBinding : public AdapterBinding
```

J#

```
public class SqlAdapterBinding extends AdapterBinding
```

JScript

```
public class SqlAdapterBinding extends AdapterBinding
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.Binding](#)

[Microsoft.ServiceModel.Channels.Common.AdapterBinding](#)

Microsoft.Adapters.Sql.SqlAdapterBinding

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Members

Represents the SQL adapter binding.

The following tables list the members exposed by the [SqlAdapterBinding](#) type.

Public Constructors

Name	Description
SqlAdapterBinding	Overloaded. Creates a new instance of the SqlAdapterBinding class.

Top

Public Properties

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 CloseTimeout	(inherited from Binding)
 ConnectionUriType	Overridden. Gets or sets the connection URI type.
 DefaultXsdFileNamePrefix	Overridden. Gets or sets the default XSD file name prefix.
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 MessageVersion	(inherited from Binding)
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 OpenTimeout	(inherited from Binding)
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Specifies the scheme name.
 SendTimeout	(inherited from Binding)
 SupportsCertificateAuthentication	Specifies whether the adapter supports certificate authentication.(inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Specifies whether the SQL adapter supports metadata browsing.
 SupportsMetadataGet	Overridden. Specifies whether the adapter supports the exposing the metadata through a GET request.

 SupportsMetadataSearch	Overridden. Specifies whether the adapter supports a metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)
 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates the binding elements.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SqlAdapterBinding Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Constructor

Creates a new instance of the SqlAdapterBinding class.

Overload List

Name	Description
SqlAdapterBinding ()	Creates a new instance of the SQLAdapterBinding class, using the default parameters.
SqlAdapterBinding (String)	Creates a new instance of the SQLAdapterBinding class, using the specified configuration name.

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Constructor ()

Creates a new instance of the SQLAdapterBinding class, using the default parameters.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<SuppressMessageAttribute("Microsoft.Performance", "CA1804:RemoveUnusedLocals", MessageId="
dummyInstance")> _
<SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="
Microsoft.Adapters.Sql.SqlAdapterBindingElement")> _
Public Sub New
```

C#

```
[SuppressMessageAttribute("Microsoft.Performance", "CA1804:RemoveUnusedLocals", MessageId="
dummyInstance")]
[SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="
Microsoft.Adapters.Sql.SqlAdapterBindingElement")]
public SqlAdapterBinding ()
```

C++

```
[SuppressMessageAttribute(L"Microsoft.Performance", L"CA1804:RemoveUnusedLocals", MessageId
=L"dummyInstance")]
[SuppressMessageAttribute(L"Microsoft.Usage", L"CA1806:DoNotIgnoreMethodResults", MessageId
=L"Microsoft.Adapters.Sql.SqlAdapterBindingElement")]
public:
SqlAdapterBinding ()
```

J#

```
/** @attribute SuppressMessageAttribute("Microsoft.Performance", "CA1804:RemoveUnusedLocals
", MessageId="dummyInstance") */
/** @attribute SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults
", MessageId="Microsoft.Adapters.Sql.SqlAdapterBindingElement") */
public SqlAdapterBinding ()
```

JScript

```
SuppressMessageAttribute("Microsoft.Performance", "CA1804:RemoveUnusedLocals", MessageId="d
ummyInstance")
SuppressMessageAttribute("Microsoft.Usage", "CA1806:DoNotIgnoreMethodResults", MessageId="M
icrosoft.Adapters.Sql.SqlAdapterBindingElement")
public function SqlAdapterBinding ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Constructor (String)

Creates a new instance of the SQLAdapterBinding class, using the specified configuration name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    configurationName As String _  
)
```

C#

```
public SqlAdapterBinding (  
    string configurationName  
)
```

C++

```
public:  
SqlAdapterBinding (  
    String^ configurationName  
)
```

J#

```
public SqlAdapterBinding (  
    String configurationName  
)
```

JScript

```
public function SqlAdapterBinding (  
    configurationName : String  
)
```

Parameters

configurationName

The configuration name of the service contract to use.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	Overloaded. (inherited from Binding)
	BuildChannelListener	Overloaded. (inherited from Binding)
	CanBuildChannelFactory	Overloaded. (inherited from Binding)
	CanBuildChannelListener	Overloaded. (inherited from Binding)
	CreateBindingElements	Overridden. Creates the binding elements.
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from Binding)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SqlAdapterBinding Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.CreateBindingElements Method

Creates the binding elements.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Function CreateBindingElements As BindingElementCollection
```

C#

```
public override BindingElementCollection CreateBindingElements ()
```

C++

```
public:  
virtual BindingElementCollection^ CreateBindingElements () override
```

J#

```
public BindingElementCollection CreateBindingElements ()
```

JScript

```
public override function CreateBindingElements () : BindingElementCollection
```

Return Value

Returns a [BindingElementCollection](#) containing the binding elements..

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding Properties

Public Properties

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 AssuresOrderedDelivery	(inherited from AdapterBinding)
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 CloseTimeout	(inherited from Binding)
 ConnectionUriType	Overridden. Gets or sets the connection URI type.
 DefaultXsdFileNamePrefix	Overridden. Gets or sets the default XSD file name prefix.
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 MessageVersion	(inherited from Binding)
 Name	(inherited from Binding)
 Namespace	(inherited from Binding)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 OpenTimeout	(inherited from Binding)
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 QueuedDelivery	(inherited from AdapterBinding)
 ReceiveTimeout	(inherited from Binding)
 Scheme	Overridden. Specifies the scheme name.
 SendTimeout	(inherited from Binding)
 SupportsCertificateAuthentication	Specifies whether the adapter supports certificate authentication. (inherited from AdapterBinding)
 SupportsMetadataBrowse	Overridden. Specifies whether the SQL adapter supports metadata browsing.
 SupportsMetadataGet	Overridden. Specifies whether the adapter supports the exposing the metadata through a GET request.
 SupportsMetadataSearch	Overridden. Specifies whether the adapter supports a metadata search.
 SupportsUserNameAuthentication	(inherited from AdapterBinding)

 SupportsWindowsAuthentication	(inherited from AdapterBinding)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

See Also

Reference

[SqlAdapterBinding Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.AllowIdentityInsert Property

Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(False, True)> _
<CategoryAttribute("Miscellaneous")> _
Public Property AllowIdentityInsert As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute("Miscellaneous")]
public bool AllowIdentityInsert { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute(L"Miscellaneous")]
public:
virtual property bool AllowIdentityInsert {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AllowIdentityInsert ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AllowIdentityInsert (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AllowIdentityInsert () : boolean

CompilerGeneratedAttribute
public final function set AllowIdentityInsert (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if you can insert values for identity columns. The default is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.BatchSize Property

Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(False, True)> _
<CategoryAttribute("Buffering")> _
Public Property BatchSize As Integer
```

C#

```
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute("Buffering")]
public int BatchSize { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute(L"Buffering")]
public:
virtual property int BatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final int get_BatchSize ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_BatchSize (int value)
```

JScript

```
CompilerGeneratedAttribute
public final function get BatchSize () : int

CompilerGeneratedAttribute
public final function set BatchSize (value : int)
```

Property Value

Returns [Int32](#).

Remarks

The default is two.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.ChunkSize Property

Specifies the buffer size used for Set<column_name> operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Buffering")> _
<BindingPropertyDirectionAttribute(False, True)> _
Public Property ChunkSize As Integer
```

C#

```
[CategoryAttribute("Buffering")]
[BindingPropertyDirectionAttribute(false, true)]
public int ChunkSize { get; set; }
```

C++

```
[CategoryAttribute(L"Buffering")]
[BindingPropertyDirectionAttribute(false, true)]
public:
virtual property int ChunkSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final int get_ChunkSize ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_ChunkSize (int value)
```

JScript

```
CompilerGeneratedAttribute
public final function get ChunkSize () : int

CompilerGeneratedAttribute
public final function set ChunkSize (value : int)
```

Property Value

Returns a [Int32](#) containing the buffer size.

Remarks

The default is 419304 bytes. A higher value may improve performance, but affects memory consumption.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.ConnectionUriType Property

Gets or sets the connection URI type.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property ConnectionUriType As Type
```

C#

```
public override Type ConnectionUriType { get; }
```

C++

```
public:  
virtual property Type^ ConnectionUriType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_ConnectionUriType ()
```

JScript

```
public override function get ConnectionUriType () : Type
```

Property Value

Returns a [Type](#) containing the URI type.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.DefaultXsdFileNamePrefix Property

Gets or sets the default XSD file name prefix.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property DefaultXsdFileNamePrefix As String
```

C#

```
public override string DefaultXsdFileNamePrefix { get; }
```

C++

```
public:  
virtual property String^ DefaultXsdFileNamePrefix {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_DefaultXsdFileNamePrefix ()
```

JScript

```
public override function get DefaultXsdFileNamePrefix () : String
```

Property Value

Returns a [String](#) containing the default prefix.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.EnableBizTalkCompatibilityMode Property

Gets or sets a value indicating if the BizTalk compatibility mode has been set.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, True)> _
<CategoryAttribute("BizTalk")> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute("BizTalk")]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute(L"BizTalk")]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnableBizTalkCompatibilityMode () : boolean

CompilerGeneratedAttribute
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the compatibility mode has been set.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.EnablePerformanceCounters Property

Gets or sets a value specifying whether to enable the WCF LOB Adpater SDK performance counters and the SQL adapter LOB latency performance counter.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, True)> _
<CategoryAttribute("Diagnostics")> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute("Diagnostics")]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute(L"Diagnostics")]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_EnablePerformanceCounters ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get EnablePerformanceCounters () : boolean

CompilerGeneratedAttribute
public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

Returns a [Boolean](#), indicating if the performance counters have been enabled.

Remarks

The default value is false; performance counters are disabled. The LOB latency performance counter measures the total time spent by the SQL adapter in making calls to the SQL server database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.Encrypt Property

Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property Encrypt As Boolean
```

C#

```
[CategoryAttribute("Connection")]
public bool Encrypt { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property bool Encrypt {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_Encrypt ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_Encrypt (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get Encrypt () : boolean

CompilerGeneratedAttribute
public final function set Encrypt (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if SQL server uses SSL encryption; the default value is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.InboundOperationType Property

Specifies if you want to perform a polling, typed polling, or notification operation.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Inbound")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[CategoryAttribute("Inbound")]
[BindingPropertyDirectionAttribute(true, false)]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[CategoryAttribute(L"Inbound")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

Returns a [InboundOperation](#) containing the operation type; the default is Polling.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.MaxConnectionPoolSize Property

Specifies the maximum number of connections allowed in a connection pool for a specific connection string.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property MaxConnectionPoolSize As Integer
```

C#

```
[CategoryAttribute("Connection")]
public int MaxConnectionPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property int MaxConnectionPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final int get_MaxConnectionPoolSize ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_MaxConnectionPoolSize (int value)
```

JScript

```
CompilerGeneratedAttribute
public final function get MaxConnectionPoolSize () : int

CompilerGeneratedAttribute
public final function set MaxConnectionPoolSize (value : int)
```

Property Value

Returns a [Int32](#) specifying the maximum number of connections; the default is 100.

Remarks

This property is used for performance tuning.

Use this property judiciously. If you set the value too large, you may exhaust the number of connections available.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.NotificationStatement Property

Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification (Inbound)")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property NotificationStatement As String
```

C#

```
[CategoryAttribute("Notification (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
public string NotificationStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Notification (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.NotifyOnListenerStart Property

Specifies whether the adapter sends a notification message when the listener is started.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Notification (Inbound)")> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Notification (Inbound)")]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Notification (Inbound)")]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the adapter sends the notification message.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.PolledDataAvailableStatement Property

Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
<BindingPropertyDirectionAttribute(True, False)> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the SQL statement specified for the PollingStatement binding property. If the first cell of the result contains a positive value, the adapter executed the polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.PollingIntervalInSeconds Property

Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling (Inbound)")> _
Public Property PollingIntervalInSeconds As Integer
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling (Inbound)")]
public int PollingIntervalInSeconds { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property int PollingIntervalInSeconds {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final int get_PollingIntervalInSeconds ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingIntervalInSeconds (int value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingIntervalInSeconds () : int

CompilerGeneratedAttribute
public final function set PollingIntervalInSeconds (value : int)
```

Property Value

Returns a [Int32](#) containing the interval.

Remarks

The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter is inactive for the remaining time in the interval.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.PollingStatement Property

Specifies the SQL statement to poll a SQL Server database table.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling (Inbound)")> _
Public Property PollingStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling (Inbound)")]
public string PollingStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property String^ PollingStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingStatement () : String

CompilerGeneratedAttribute
public final function set PollingStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

You can specify a simple SELECT statement or stored procedures for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.

You can specify any number of SQL statements separated by a semicolon. You can use the polling statement to read or update data in a SQL Server database table. The SQL adapter executes the polling statements inside one transaction. When the adapter is used with BizTalk Server, the same transaction is used to submit messages from SQL Server to the BizTalk message box.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.PollWhileDataFound Property

Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling (Inbound)")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling (Inbound)")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

Returns a [Boolean](#) specifying whether to ignore the polling interval; the default is false.

Remarks

If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval.

Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for `PolledDataAvailableStatement` returns that data is available for polling. The adapter then executes the statement specified for the `PollingStatement` binding property. Assuming that the adapter takes just 10 seconds to execute the polling statement, it will now have to wait for 50 seconds before executing the `PolledDataAvailableStatement` again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the `PollWhileDataFound` binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.Scheme Property

Specifies the scheme name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property Scheme As String
```

C#

```
public override string Scheme { get; }
```

C++

```
public:  
virtual property String^ Scheme {  
    String^ get () override;  
}
```

J#

```
/** @property */  
public String get_Scheme ()
```

JScript

```
public override function get Scheme () : String
```

Property Value

Returns a [String](#) containing the scheme name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.SupportsMetadataBrowse Property

Specifies whether the SQL adapter supports metadata browsing.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataBrowse As Boolean
```

C#

```
public override bool SupportsMetadataBrowse { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataBrowse {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataBrowse ()
```

JScript

```
public override function get SupportsMetadataBrowse () : boolean
```

Property Value

Returns a [Boolean](#) indicating if the adapter supports metadata browsing.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.SupportsMetadataGet Property

Specifies whether the adapter supports the exposing the metadata through a GET request.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property SupportsMetadataGet As Boolean
```

C#

```
public override bool SupportsMetadataGet { get; }
```

C++

```
public:  
virtual property bool SupportsMetadataGet {  
    bool get () override;  
}
```

J#

```
/** @property */  
public boolean get_SupportsMetadataGet ()
```

JScript

```
public override function get SupportsMetadataGet () : boolean
```

Property Value

Returns a [Boolean](#) indicating if the metadata has been exposed to a GET request.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.SupportsMetadataSearch Property

Specifies whether the adapter supports a metadata search.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public Overrides ReadOnly Property SupportsMetadataSearch As Boolean
```

C#

```
public override bool SupportsMetadataSearch { get; }
```

C++

```
public:
virtual property bool SupportsMetadataSearch {
    bool get () override;
}
```

J#

```
/** @property */
public boolean get_SupportsMetadataSearch ()
```

JScript

```
public override function get SupportsMetadataSearch () : boolean
```

Property Value

Returns a [Boolean](#) indicating if the adapter supports a metadata search.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.UseAmbientTransaction Property

Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Transaction")> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[CategoryAttribute("Transaction")]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[CategoryAttribute(L"Transaction")]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseAmbientTransaction ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseAmbientTransaction () : boolean

CompilerGeneratedAttribute
public final function set UseAmbientTransaction (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the SQL adapter performs operations using a transactional context; the default is true. Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.UseDatabaseNameInXsdNamespace Property

Specifies whether the XSD generated for a particular artifact contains the database name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, True)> _
<CategoryAttribute("Metadata")> _
Public Property UseDatabaseNameInXsdNamespace As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute("Metadata")]
public bool UseDatabaseNameInXsdNamespace { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, true)]
[CategoryAttribute(L"Metadata")]
public:
virtual property bool UseDatabaseNameInXsdNamespace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseDatabaseNameInXsdNamespace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseDatabaseNameInXsdNamespace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseDatabaseNameInXsdNamespace () : boolean

CompilerGeneratedAttribute
public final function set UseDatabaseNameInXsdNamespace (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the XSD contains the database name; the default is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.WorkstationId Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property WorkstationId As String
```

C#

```
[CategoryAttribute("Connection")]
public string WorkstationId { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property String^ WorkstationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_WorkstationId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_WorkstationId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get WorkstationId () : String

CompilerGeneratedAttribute
public final function set WorkstationId (value : String)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.XmlStoredProcedureRootNodeName Property

Specifies the XML root node name of the stored procedures for the database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("FOR XML")> _
Public Property XmlStoredProcedureRootNodeName As String
```

C#

```
[CategoryAttribute("FOR XML")]
public string XmlStoredProcedureRootNodeName { get; set; }
```

C++

```
[CategoryAttribute(L"FOR XML")]
public:
virtual property String^ XmlStoredProcedureRootNodeName {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_XmlStoredProcedureRootNodeName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_XmlStoredProcedureRootNodeName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get XmlStoredProcedureRootNodeName () : String

CompilerGeneratedAttribute
public final function set XmlStoredProcedureRootNodeName (value : String)
```

Property Value

Returns a [String](#) containing the root node name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBinding.XmlStoredProcedureRootNodeNamespace Property

Specifies the XML stored procedure root node namespace.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("FOR XML")> _
Public Property XmlStoredProcedureRootNodeNamespace As String
```

C#

```
[CategoryAttribute("FOR XML")]
public string XmlStoredProcedureRootNodeNamespace { get; set; }
```

C++

```
[CategoryAttribute(L"FOR XML")]
public:
virtual property String^ XmlStoredProcedureRootNodeNamespace {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_XmlStoredProcedureRootNodeNamespace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_XmlStoredProcedureRootNodeNamespace (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get XmlStoredProcedureRootNodeNamespace () : String

CompilerGeneratedAttribute
public final function set XmlStoredProcedureRootNodeNamespace (value : String)
```

Property Value

Returns a [String](#) containing the namespace name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBinding Class](#)

[SqlAdapterBinding Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingCollectionElement Class

Describes an element of a SQL adapter binding collection.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class SqlAdapterBindingCollectionElement
    Inherits StandardBindingCollectionElement(Of SqlAdapterBinding, SqlAdapterBindingConfigura
tionElement)
```

C#

```
public sealed class SqlAdapterBindingCollectionElement : StandardBindingCollectionElement<S
qlAdapterBinding, SqlAdapterBindingConfigurationElement>
```

C++

```
public ref class SqlAdapterBindingCollectionElement sealed : public StandardBindingCollecti
onElement<SqlAdapterBinding^, SqlAdapterBindingConfigurationElement^>
```

J#

```
public final class SqlAdapterBindingCollectionElement extends StandardBindingCollectionElem
ent<SqlAdapterBinding, SqlAdapterBindingConfigurationElement>
```

JScript

```
public final class SqlAdapterBindingCollectionElement extends StandardBindingCollectionElem
ent<SqlAdapterBinding, SqlAdapterBindingConfigurationElement>
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.BindingCollectionElement](#)

[System.ServiceModel.Configuration.StandardBindingCollectionElement](#)

Microsoft.Adapters.Sql.SqlAdapterBindingCollectionElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingCollectionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingCollectionElement Members

Describes an element of a SQL adapter binding collection.

The following tables list the members exposed by the [SqlAdapterBindingCollectionElement](#) type.

Public Constructors

Name	Description
SqlAdapterBindingCollectionElement	creates a new instance of the SqlAdapterBindingCollectionElement class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BindingName	(inherited from BindingCollectionElement)
 Bindings	(inherited from StandardBindingCollectionElement)
 BindingType	(inherited from StandardBindingCollectionElement)
 ConfiguredBindings	(inherited from StandardBindingCollectionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from StandardBindingCollectionElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 ContainsKey	(inherited from StandardBindingCollectionElement)
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 GetDefault	(inherited from StandardBindingCollectionElement)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ConfigurationElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)

 Reset	(inherited from ConfigurationElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ConfigurationElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 TryAdd	(inherited from StandardBindingCollectionElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[SqlAdapterBindingCollectionElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingCollectionElement Constructor

creates a new instance of the SqlAdapterBindingCollectionElement class.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterBindingCollectionElement ()
```

C++

```
public:  
SqlAdapterBindingCollectionElement ()
```

J#

```
public SqlAdapterBindingCollectionElement ()
```

JScript

```
public function SqlAdapterBindingCollectionElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingCollectionElement Class](#)

[SqlAdapterBindingCollectionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingCollectionElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ContainsKey	(inherited from StandardBindingCollectionElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	GetDefault	(inherited from StandardBindingCollectionElement)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ConfigurationElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	TryAdd	(inherited from StandardBindingCollectionElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[SqlAdapterBindingCollectionElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingCollectionElement Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	BindingName	(inherited from BindingCollectionElement)
	Bindings	(inherited from StandardBindingCollectionElement)
	BindingType	(inherited from StandardBindingCollectionElement)
	ConfiguredBindings	(inherited from StandardBindingCollectionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingCollectionElement)

See Also

Reference

[SqlAdapterBindingCollectionElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement Class

Describes a SQL adapter binding configuration element.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class SqlAdapterBindingConfigurationElement
    Inherits StandardBindingElement
```

C#

```
public sealed class SqlAdapterBindingConfigurationElement : StandardBindingElement
```

C++

```
public ref class SqlAdapterBindingConfigurationElement sealed : public StandardBindingElement
```

J#

```
public final class SqlAdapterBindingConfigurationElement extends StandardBindingElement
```

JScript

```
public final class SqlAdapterBindingConfigurationElement extends StandardBindingElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.StandardBindingElement](#)

Microsoft.Adapters.Sql.SqlAdapterBindingConfigurationElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Members](#)

SqlAdapterBindingConfigurationElement Members

Describes a SQL adapter binding configuration element.

The following tables list the members exposed by the [SqlAdapterBindingConfigurationElement](#) type.

Public Constructors

Name	Description
SqlAdapterBindingConfigurationElement	Creates a new instance of the SqlAdapterBindingConfigurationElement class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Specifies whether the adapter can insert values for identity columns during Insert and Update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	
 CloseTimeout	(inherited from StandardBindingElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 Name	(inherited from StandardBindingElement)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 OpenTimeout	(inherited from StandardBindingElement)
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.

 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.
---	---

[Top](#)

Protected Properties

	Name	Description
	BindingElementType	(inherited from StandardBindingElement)
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from StandardBindingElement)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from StandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnApplyConfiguration	(inherited from StandardBindingElement)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from StandardBindingElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement Constructor

Creates a new instance of the SqlAdapterBindingConfigurationElement class.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterBindingConfigurationElement ()
```

C++

```
public:  
SqlAdapterBindingConfigurationElement ()
```

J#

```
public SqlAdapterBindingConfigurationElement ()
```

JScript

```
public function SqlAdapterBindingConfigurationElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	(inherited from StandardBindingElement)
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from StandardBindingElement)
	IsModified	(inherited from ConfigurationElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnApplyConfiguration	(inherited from StandardBindingElement)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from StandardBindingElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ConfigurationElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Specifies whether the adapter can insert values for identity columns during Insert and Update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	
 CloseTimeout	(inherited from StandardBindingElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 Name	(inherited from StandardBindingElement)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 OpenTimeout	(inherited from StandardBindingElement)
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 ReceiveTimeout	(inherited from StandardBindingElement)
 SendTimeout	(inherited from StandardBindingElement)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

Protected Properties

Name	Description
 BindingElementType	(inherited from StandardBindingElement)
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)

 Properties	(inherited from StandardBindingElement)
--	--

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.AllowIdentityInsert Property

Specifies whether the adapter can insert values for identity columns during Insert and Update operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("allowIdentityInsert", DefaultValue:=False)> _
<CategoryAttribute("Miscellaneous")> _
<BindingPropertyDirectionAttribute(False, True)> _
Public Property AllowIdentityInsert As Boolean
```

C#

```
[ConfigurationPropertyAttribute("allowIdentityInsert", DefaultValue=false)]
[CategoryAttribute("Miscellaneous")]
[BindingPropertyDirectionAttribute(false, true)]
public bool AllowIdentityInsert { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"allowIdentityInsert", DefaultValue=false)]
[CategoryAttribute(L"Miscellaneous")]
[BindingPropertyDirectionAttribute(false, true)]
public:
virtual property bool AllowIdentityInsert {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AllowIdentityInsert ()

/** @property */
public final void set_AllowIdentityInsert (boolean value)
```

JScript

```
public final function get AllowIdentityInsert () : boolean

public final function set AllowIdentityInsert (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if you can insert values for identity columns. The default is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.BatchSize Property

Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Buffering")> _
<ConfigurationPropertyAttribute("batchSize", DefaultValue:=20)> _
<BindingPropertyDirectionAttribute(False, True)> _
Public Property BatchSize As Integer
```

C#

```
[CategoryAttribute("Buffering")]
[ConfigurationPropertyAttribute("batchSize", DefaultValue=20)]
[BindingPropertyDirectionAttribute(false, true)]
public int BatchSize { get; set; }
```

C++

```
[CategoryAttribute(L"Buffering")]
[ConfigurationPropertyAttribute(L"batchSize", DefaultValue=20)]
[BindingPropertyDirectionAttribute(false, true)]
public:
virtual property int BatchSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_BatchSize ()

/** @property */
public final void set_BatchSize (int value)
```

JScript

```
public final function get BatchSize () : int

public final function set BatchSize (value : int)
```

Property Value

Returns a [Int32](#) indicating the batch size.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.ChunkSize Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("chunkSize", DefaultValue:=4194304)> _
<BindingPropertyDirectionAttribute(False, True)> _
<CategoryAttribute("Buffering")> _
Public Property ChunkSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("chunkSize", DefaultValue=4194304)]
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute("Buffering")]
public int ChunkSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"chunkSize", DefaultValue=4194304)]
[BindingPropertyDirectionAttribute(false, true)]
[CategoryAttribute(L"Buffering")]
public:
virtual property int ChunkSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ChunkSize ()

/** @property */
public final void set_ChunkSize (int value)
```

JScript

```
public final function get ChunkSize () : int

public final function set ChunkSize (value : int)
```

Property Value

Returns [Int32](#).

Remarks

The default is two.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.EnableBizTalkCompatibilityMode Property

Gets or sets a value indicating if the BizTalk compatibility mode has been set.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("BizTalk")> _
<ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue:=True)> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[CategoryAttribute("BizTalk")]
[ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue=true)]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[CategoryAttribute(L"BizTalk")]
[ConfigurationPropertyAttribute(L"enableBizTalkCompatibilityMode", DefaultValue=true)]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean

public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the compatibility mode has been set.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)
[SqlAdapterBindingConfigurationElement Members](#)
[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.EnablePerformanceCounters Property

Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("Diagnostics")> _
<ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue:=False)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue=false)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[ConfigurationPropertyAttribute(L"enablePerformanceCounters", DefaultValue=false)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

Returns a [Boolean](#), indicating if the performance counters have been enabled.

Remarks

The default value is false; performance counters are disabled. The LOB latency performance counter measures the total time spent by the SQL adapter in making calls to the SQL server database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.Encrypt Property

Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("encrypt", DefaultValue:=False)> _
<CategoryAttribute("Connection")> _
Public Property Encrypt As Boolean
```

C#

```
[ConfigurationPropertyAttribute("encrypt", DefaultValue=false)]
[CategoryAttribute("Connection")]
public bool Encrypt { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"encrypt", DefaultValue=false)]
[CategoryAttribute(L"Connection")]
public:
virtual property bool Encrypt {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_Encrypt ()

/** @property */
public final void set_Encrypt (boolean value)
```

JScript

```
public final function get Encrypt () : boolean

public final function set Encrypt (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if SQL server uses SSL encryption; the default value is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.InboundOperationType Property

Specifies if you want to perform a polling, typed polling, or notification operation.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("inboundOperationType", DefaultValue:=InboundOperation.Polling)
> _
<CategoryAttribute("Inbound")> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("inboundOperationType", DefaultValue=InboundOperation.Polling)
]
[CategoryAttribute("Inbound")]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"inboundOperationType", DefaultValue=InboundOperation::Polling)
]
[CategoryAttribute(L"Inbound")]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
public final InboundOperation get_InboundOperationType ()

/** @property */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
public final function get InboundOperationType () : InboundOperation
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

Returns a [InboundOperation](#) containing the operation type; the default is Polling.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.MaxConnectionPoolSize Property

Specifies the maximum number of connections allowed in a connection pool for a specific connection string.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Connection")> _
<ConfigurationPropertyAttribute("maxConnectionPoolSize", DefaultValue:=100)> _
Public Property MaxConnectionPoolSize As Integer
```

C#

```
[CategoryAttribute("Connection")]
[ConfigurationPropertyAttribute("maxConnectionPoolSize", DefaultValue=100)]
public int MaxConnectionPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
[ConfigurationPropertyAttribute(L"maxConnectionPoolSize", DefaultValue=100)]
public:
virtual property int MaxConnectionPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionPoolSize ()

/** @property */
public final void set_MaxConnectionPoolSize (int value)
```

JScript

```
public final function get MaxConnectionPoolSize () : int

public final function set MaxConnectionPoolSize (value : int)
```

Property Value

Returns a [Int32](#) specifying the maximum number of connections; the default is 100.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.NotificationStatement Property

Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Notification (Inbound)")> _
<ConfigurationPropertyAttribute("notificationStatement", DefaultValue:=<string>)> _
Public Property NotificationStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Notification (Inbound)")]
[ConfigurationPropertyAttribute("notificationStatement", DefaultValue="")]
public string NotificationStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Notification (Inbound)")]
[ConfigurationPropertyAttribute(L"notificationStatement", DefaultValue=L"")]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NotificationStatement ()

/** @property */
public final void set_NotificationStatement (String value)
```

JScript

```
public final function get NotificationStatement () : String
public final function set NotificationStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.NotifyOnListenerStart Property

Specifies whether the adapter sends a notification message when the listener is started.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification (Inbound)")> _
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue:=True)> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[CategoryAttribute("Notification (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue=true)]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[CategoryAttribute(L"Notification (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"notifyOnListenerStart", DefaultValue=true)]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NotifyOnListenerStart ()

/** @property */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
public final function get NotifyOnListenerStart () : boolean

public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the adapter sends the notification message.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not

guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.PolledDataAvailableStatement Property

Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<CategoryAttribute("Polling (Inbound)")> _
<ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue:= "")> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute("Polling (Inbound)")]
[ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue="")]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[CategoryAttribute(L"Polling (Inbound)")]
[ConfigurationPropertyAttribute(L"polledDataAvailableStatement", DefaultValue=L"")]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PolledDataAvailableStatement ()

/** @property */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
public final function get PolledDataAvailableStatement () : String

public final function set PolledDataAvailableStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the SQL statement specified for the [PollingStatement](#) binding property. If the first cell of the result contains a positive value, the adapter executed the polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.PollingIntervalInSeconds Property

Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Polling (Inbound)")> _
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingIntervalInSeconds", DefaultValue:=30)> _
Public Property PollingIntervalInSeconds As Integer
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingIntervalInSeconds", DefaultValue=30)]
public int PollingIntervalInSeconds { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingIntervalInSeconds", DefaultValue=30)]
public:
virtual property int PollingIntervalInSeconds {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingIntervalInSeconds ()

/** @property */
public final void set_PollingIntervalInSeconds (int value)
```

JScript

```
public final function get PollingIntervalInSeconds () : int
public final function set PollingIntervalInSeconds (value : int)
```

Property Value

Returns a [Int32](#) containing the interval.

Remarks

The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter is inactive for the remaining time in the interval.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.PollingStatement Property

Specifies the SQL statement to poll a SQL Server database table.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollingStatement", DefaultValue:= "")> _
Public Property PollingStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollingStatement", DefaultValue="")]
public string PollingStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollingStatement", DefaultValue=L"")]
public:
virtual property String^ PollingStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingStatement ()

/** @property */
public final void set_PollingStatement (String value)
```

JScript

```
public final function get PollingStatement () : String

public final function set PollingStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

You can specify a simple SELECT statement or stored procedures for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.

You can specify any number of SQL statements separated by a semicolon. You can use the polling statement to read or update data in a SQL Server database table. The SQL adapter executes the polling statements inside one transaction. When the adapter is used with BizTalk Server, the same transaction is used to submit messages from SQL Server to the BizTalk message box.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.PollWhileDataFound Property

Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<BindingPropertyDirectionAttribute(True, False)> _
<ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue:=False)> _
<CategoryAttribute("Polling (Inbound)")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue=false)]
[CategoryAttribute("Polling (Inbound)")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[BindingPropertyDirectionAttribute(true, false)]
[ConfigurationPropertyAttribute(L"pollWhileDataFound", DefaultValue=false)]
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_PollWhileDataFound ()

/** @property */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
public final function get PollWhileDataFound () : boolean

public final function set PollWhileDataFound (value : boolean)
```

Property Value

Returns a [Boolean](#) specifying whether to ignore the polling interval; the default is false.

Remarks

If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval.

Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for `PolledDataAvailableStatement` returns that data is available for polling. The adapter then executes the statement specified for the `PollingStatement` binding property. Assuming that the adapter takes just 10 seconds to execute the polling statement, it will now have to wait for 50 seconds before executing the `PolledDataAvailableStatement` again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the `PollWhileDataFound` binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.UseAmbientTransaction Property

Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue:=True)> _
<CategoryAttribute("Transaction")> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue=true)]
[CategoryAttribute("Transaction")]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useAmbientTransaction", DefaultValue=true)]
[CategoryAttribute(L"Transaction")]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the SQL adapter performs operations using a transactional context; the default is true. Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.UseDatabaseNameInXsdNamespace Property

Specifies whether the XSD generated for a particular artifact contains the database name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Metadata")> _
<ConfigurationPropertyAttribute("useDatabaseNameInXsdNamespace", DefaultValue:=False)> _
Public Property UseDatabaseNameInXsdNamespace As Boolean
```

C#

```
[CategoryAttribute("Metadata")]
[ConfigurationPropertyAttribute("useDatabaseNameInXsdNamespace", DefaultValue=false)]
public bool UseDatabaseNameInXsdNamespace { get; set; }
```

C++

```
[CategoryAttribute(L"Metadata")]
[ConfigurationPropertyAttribute(L"useDatabaseNameInXsdNamespace", DefaultValue=false)]
public:
virtual property bool UseDatabaseNameInXsdNamespace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseDatabaseNameInXsdNamespace ()

/** @property */
public final void set_UseDatabaseNameInXsdNamespace (boolean value)
```

JScript

```
public final function get UseDatabaseNameInXsdNamespace () : boolean

public final function set UseDatabaseNameInXsdNamespace (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the XSD contains the database name; the default is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.WorkstationId Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
<ConfigurationPropertyAttribute("workstationId", DefaultValue:= "")> _
Public Property WorkstationId As String
```

C#

```
[CategoryAttribute("Connection")]
[ConfigurationPropertyAttribute("workstationId", DefaultValue="")]
public string WorkstationId { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
[ConfigurationPropertyAttribute(L"workstationId", DefaultValue=L"")]
public:
virtual property String^ WorkstationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_WorkstationId ()

/** @property */
public final void set_WorkstationId (String value)
```

JScript

```
public final function get WorkstationId () : String

public final function set WorkstationId (value : String)
```

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.XmlStoredProcedureRootNodeName Property

Specifies the XML root node name of the stored procedures for the database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("FOR XML")> _
<ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeName", DefaultValue="")> _
Public Property XmlStoredProcedureRootNodeName As String
```

C#

```
[CategoryAttribute("FOR XML")]
[ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeName", DefaultValue="")]
public string XmlStoredProcedureRootNodeName { get; set; }
```

C++

```
[CategoryAttribute(L"FOR XML")]
[ConfigurationPropertyAttribute(L"xmlStoredProcedureRootNodeName", DefaultValue=L"")]
public:
virtual property String^ XmlStoredProcedureRootNodeName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_XmlStoredProcedureRootNodeName ()

/** @property */
public final void set_XmlStoredProcedureRootNodeName (String value)
```

JScript

```
public final function get XmlStoredProcedureRootNodeName () : String
public final function set XmlStoredProcedureRootNodeName (value : String)
```

Property Value

Returns a [String](#) containing the root node name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingConfigurationElement.XmlStoredProcedureRootNodeNamespace Property

Specifies the XML stored procedure root node namespace.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("FOR XML")> _
<ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeNamespace", DefaultValue="")> _
Public Property XmlStoredProcedureRootNodeNamespace As String
```

C#

```
[CategoryAttribute("FOR XML")]
[ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeNamespace", DefaultValue="")]
public string XmlStoredProcedureRootNodeNamespace { get; set; }
```

C++

```
[CategoryAttribute(L"FOR XML")]
[ConfigurationPropertyAttribute(L"xmlStoredProcedureRootNodeNamespace", DefaultValue=L"")]
public:
virtual property String^ XmlStoredProcedureRootNodeNamespace {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_XmlStoredProcedureRootNodeNamespace ()

/** @property */
public final void set_XmlStoredProcedureRootNodeNamespace (String value)
```

JScript

```
public final function get XmlStoredProcedureRootNodeNamespace () : String

public final function set XmlStoredProcedureRootNodeNamespace (value : String)
```

Property Value

Returns a [String](#) containing the namespace name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingConfigurationElement Class](#)

[SqlAdapterBindingConfigurationElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Class

Describes a SQL adapter binding element.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SqlAdapterBindingElement  
    Inherits CommonAdapter
```

C#

```
public sealed class SqlAdapterBindingElement : CommonAdapter
```

C++

```
public ref class SqlAdapterBindingElement sealed : public CommonAdapter
```

J#

```
public final class SqlAdapterBindingElement extends CommonAdapter
```

JScript

```
public final class SqlAdapterBindingElement extends CommonAdapter
```

Inheritance Hierarchy [System.Object](#)

[System.ServiceModel.Channels.BindingElement](#)

[System.ServiceModel.Channels.TransportBindingElement](#)

[Microsoft.ServiceModel.Channels.Common.Adapter](#)

[Microsoft.Adapters.AdapterUtilities.CommonAdapter](#)

Microsoft.Adapters.Sql.SqlAdapterBindingElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

SqlAdapterBindingElement Members
Microsoft.Adapters.Sql Namespace

SqlAdapterBindingElement Members

Describes a SQL adapter binding element.

The following tables list the members exposed by the [SqlAdapterBindingElement](#) type.

Public Constructors

Name	Description
SqlAdapterBindingElement	Overloaded. Creates a new instance of the SqlAdapterBindingElement class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 Scheme	Overridden. Specifies the scheme name.
 TransactedReceiveEnabled	(inherited from Adapter)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

Top

Protected Properties

Name	Description
 Namespace	(inherited from Adapter)
 Settings	(inherited from Adapter)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	BuildConnectionFactory	(inherited from Adapter)
	BuildConnectionUri	(inherited from Adapter)
	BuildWsdRetrieval	(inherited from Adapter)
	CloneAdapter	(inherited from Adapter)
	Finalize	(inherited from Object)
	IsHandlerSupported	(inherited from Adapter)
	MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SqlAdapterBindingElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Constructor

Creates a new instance of the SqlAdapterBindingElement class.

Overload List

Name	Description
SqlAdapterBindingElement ()	Creates a new instance of the SqlAdapterBindingElement class, using the default values.
SqlAdapterBindingElement (SqlAdapterBindingElement)	Creates a new instance of the SqlAdapterBindingElement class, using the values from the specified adapter.

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Constructor ()

Creates a new instance of the SqlAdapterBindingElement class, using the default values.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterBindingElement ()
```

C++

```
public:  
SqlAdapterBindingElement ()
```

J#

```
public SqlAdapterBindingElement ()
```

JScript

```
public function SqlAdapterBindingElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Constructor (SqlAdapterBindingElement)

Creates a new instance of the SqlAdapterBindingElement class, using the values from the specified adapter.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Sub New ( _  
    adapterToBeCloned As SqlAdapterBindingElement _  
)
```

C#

```
public SqlAdapterBindingElement (  
    SqlAdapterBindingElement adapterToBeCloned  
)
```

C++

```
public:  
SqlAdapterBindingElement (  
    SqlAdapterBindingElement^ adapterToBeCloned  
)
```

J#

```
public SqlAdapterBindingElement (  
    SqlAdapterBindingElement adapterToBeCloned  
)
```

JScript

```
public function SqlAdapterBindingElement (  
    adapterToBeCloned : SqlAdapterBindingElement  
)
```

Parameters

adapterToBeCloned

The adapter to clone.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	BuildChannelFactory	(inherited from Adapter)
	BuildChannelListener	(inherited from Adapter)
	CanBuildChannelFactory	(inherited from Adapter)
	CanBuildChannelListener	(inherited from Adapter)
	Clone	(inherited from Adapter)
	Equals	Overloaded. (inherited from Object)
	GetHashCode	(inherited from Object)
	GetProperty	(inherited from TransportBindingElement)
	GetType	(inherited from Object)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	BuildConnectionFactory	(inherited from Adapter)
	BuildConnectionUri	(inherited from Adapter)
	BuildWsdRetrieval	(inherited from Adapter)
	CloneAdapter	(inherited from Adapter)
	Finalize	(inherited from Object)
	IsHandlerSupported	(inherited from Adapter)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SqlAdapterBindingElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 ManualAddressing	(inherited from TransportBindingElement)
 MaxBufferPoolSize	(inherited from TransportBindingElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 MaxReceivedMessageSize	(inherited from TransportBindingElement)
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 Scheme	Overridden. Specifies the scheme name.
 TransactedReceiveEnabled	(inherited from Adapter)
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

Protected Properties

Name	Description
 Namespace	(inherited from Adapter)
 Settings	(inherited from Adapter)

See Also

Reference

[SqlAdapterBindingElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.AllowIdentityInsert Property

Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Miscellaneous")> _
Public Property AllowIdentityInsert As Boolean
```

C#

```
[CategoryAttribute("Miscellaneous")]
public bool AllowIdentityInsert { get; set; }
```

C++

```
[CategoryAttribute(L"Miscellaneous")]
public:
virtual property bool AllowIdentityInsert {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_AllowIdentityInsert ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_AllowIdentityInsert (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get AllowIdentityInsert () : boolean

CompilerGeneratedAttribute
public final function set AllowIdentityInsert (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if you can insert values for identity columns. The default is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.BatchSize Property

Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property BatchSize As Integer
```

C#

```
public int BatchSize { get; set; }
```

C++

```
public:  
virtual property int BatchSize {  
    int get () sealed;  
    void set (int value) sealed;  
}
```

J#

```
/** @property */  
public final int get_BatchSize ()  
  
/** @property */  
public final void set_BatchSize (int value)
```

JScript

```
public final function get BatchSize () : int  
  
public final function set BatchSize (value : int)
```

Property Value

Returns [Int32](#).

Remarks

The default is two.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.ChunkSize Property

Specifies the buffer size used for Set<column_name> operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public Property ChunkSize As Integer
```

C#

```
public int ChunkSize { get; set; }
```

C++

```
public:
virtual property int ChunkSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ChunkSize ()

/** @property */
public final void set_ChunkSize (int value)
```

JScript

```
public final function get ChunkSize () : int

public final function set ChunkSize (value : int)
```

Property Value

Returns a [Int32](#) containing the buffer size.

Remarks

The default is 419304 bytes. A higher value may improve performance, but affects memory consumption.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.EnableBizTalkCompatibilityMode Property

Gets or sets a value indicating if the BizTalk compatibility mode has been set.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
public:  
virtual property bool EnableBizTalkCompatibilityMode {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnableBizTalkCompatibilityMode ()  
  
/** @property */  
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean  
  
public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the compatibility mode has been set.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.EnablePerformanceCounters Property

Gets or sets a value specifying whether to enable the WCF LOB Adpater SDK performance counters and the SQL adapter LOB latency performance counter.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property EnablePerformanceCounters As Boolean
```

C#

```
public bool EnablePerformanceCounters { get; set; }
```

C++

```
public:  
virtual property bool EnablePerformanceCounters {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_EnablePerformanceCounters ()  
  
/** @property */  
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean  
  
public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

Returns a [Boolean](#), indicating if the performance counters have been enabled.

Remarks

The default value is false; performance counters are disabled. The LOB latency performance counter measures the total time spent by the SQL adapter in making calls to the SQL server database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.Encrypt Property

Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property Encrypt As Boolean
```

C#

```
[CategoryAttribute("Connection")]
public bool Encrypt { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property bool Encrypt {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_Encrypt ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_Encrypt (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get Encrypt () : boolean

CompilerGeneratedAttribute
public final function set Encrypt (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if SQL server uses SSL encryption; the default value is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.InboundOperationType Property

Specifies if you want to perform a polling, typed polling, or notification operation.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public Property InboundOperationType As InboundOperation
```

C#

```
public InboundOperation InboundOperationType { get; set; }
```

C++

```
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final InboundOperation get_InboundOperationType ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
CompilerGeneratedAttribute
public final function get InboundOperationType () : InboundOperation

CompilerGeneratedAttribute
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

Returns a [InboundOperation](#) containing the operation type; the default is Polling.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.MaxConnectionPoolSize Property

Specifies the maximum number of connections allowed in a connection pool for a specific connection string.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property MaxConnectionPoolSize As Integer
```

C#

```
[CategoryAttribute("Connection")]
public int MaxConnectionPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property int MaxConnectionPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionPoolSize ()

/** @property */
public final void set_MaxConnectionPoolSize (int value)
```

JScript

```
public final function get MaxConnectionPoolSize () : int
public final function set MaxConnectionPoolSize (value : int)
```

Property Value

Returns a [Int32](#) specifying the maximum number of connections; the default is 100.

Remarks

This property is used for performance tuning.

Use this property judiciously. If you set the value too large, you may exhaust the number of connections available.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.NotificationStatement Property

Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification (Inbound)")> _
Public Property NotificationStatement As String
```

C#

```
[CategoryAttribute("Notification (Inbound)")]
public string NotificationStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Notification (Inbound)")]
public:
virtual property String^ NotificationStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_NotificationStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotificationStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotificationStatement () : String

CompilerGeneratedAttribute
public final function set NotificationStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.NotifyOnListenerStart Property

Specifies whether the adapter sends a notification message when the listener is started.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification (Inbound)")> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[CategoryAttribute("Notification (Inbound)")]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[CategoryAttribute(L"Notification (Inbound)")]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_NotifyOnListenerStart ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get NotifyOnListenerStart () : boolean

CompilerGeneratedAttribute
public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the adapter sends the notification message.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.PolledDataAvailableStatement Property

Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PolledDataAvailableStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PolledDataAvailableStatement () : String

CompilerGeneratedAttribute
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the SQL statement specified for the PollingStatement binding property. If the first cell of the result contains a positive value, the adapter executed the polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.PollingIntervalInSeconds Property

Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
Public Property PollingIntervalInSeconds As Integer
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
public int PollingIntervalInSeconds { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property int PollingIntervalInSeconds {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingIntervalInSeconds ()

/** @property */
public final void set_PollingIntervalInSeconds (int value)
```

JScript

```
public final function get PollingIntervalInSeconds () : int

public final function set PollingIntervalInSeconds (value : int)
```

Property Value

Returns a [Int32](#) containing the interval.

Remarks

The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter is inactive for the remaining time in the interval.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.PollingStatement Property

Specifies the SQL statement to poll a SQL Server database table.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
Public Property PollingStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
public string PollingStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property String^ PollingStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_PollingStatement ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollingStatement (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollingStatement () : String

CompilerGeneratedAttribute
public final function set PollingStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

You can specify a simple SELECT statement or stored procedures for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling, which is determined by the PolledDataAvailableStatement binding property.

You can specify any number of SQL statements separated by a semicolon. You can use the polling statement to read or update data in a SQL Server database table. The SQL adapter executes the polling statements inside one transaction. When the adapter is used with BizTalk Server, the same transaction is used to submit messages from SQL Server to the BizTalk message box.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.PollWhileDataFound Property

Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_PollWhileDataFound ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get PollWhileDataFound () : boolean

CompilerGeneratedAttribute
public final function set PollWhileDataFound (value : boolean)
```

Property Value

Returns a [Boolean](#) specifying whether to ignore the polling interval; the default is false.

Remarks

If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval.

Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for

PolledDataAvailableStatement returns that data is available for polling. The adapter then executes the statement specified for the PollingStatement binding property. Assuming that the adapter takes just 10 seconds to execute the polling statement, it will now have to wait for 50 seconds before executing the PolledDataAvailableStatement again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the PollWhileDataFound binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.Scheme Property

Specifies the scheme name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides NotOverridable ReadOnly Property Scheme As String
```

C#

```
public override sealed string Scheme { get; }
```

C++

```
public:  
virtual property String^ Scheme {  
    String^ get () override sealed;  
}
```

J#

```
/** @property */  
public final String get_Scheme ()
```

JScript

```
public override final function get Scheme () : String
```

Property Value

Returns a [String](#) containing the scheme name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.UseAmbientTransaction Property

Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property UseAmbientTransaction As Boolean
```

C#

```
public bool UseAmbientTransaction { get; set; }
```

C++

```
public:  
virtual property bool UseAmbientTransaction {  
    bool get () sealed;  
    void set (bool value) sealed;  
}
```

J#

```
/** @property */  
public final boolean get_UseAmbientTransaction ()  
  
/** @property */  
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean  
  
public final function set UseAmbientTransaction (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the SQL adapter performs operations using a transactional context; the default is true. Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.UseDatabaseNameInXsdNamespace Property

Specifies whether the XSD generated for a particular artifact contains the database name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Metadata")> _
Public Property UseDatabaseNameInXsdNamespace As Boolean
```

C#

```
[CategoryAttribute("Metadata")]
public bool UseDatabaseNameInXsdNamespace { get; set; }
```

C++

```
[CategoryAttribute(L"Metadata")]
public:
virtual property bool UseDatabaseNameInXsdNamespace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final boolean get_UseDatabaseNameInXsdNamespace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_UseDatabaseNameInXsdNamespace (boolean value)
```

JScript

```
CompilerGeneratedAttribute
public final function get UseDatabaseNameInXsdNamespace () : boolean

CompilerGeneratedAttribute
public final function set UseDatabaseNameInXsdNamespace (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the XSD contains the database name; the default is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.WorkstationId Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Connection")> _
Public Property WorkstationId As String
```

C#

```
[CategoryAttribute("Connection")]
public string WorkstationId { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
public:
virtual property String^ WorkstationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_WorkstationId ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_WorkstationId (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get WorkstationId () : String

CompilerGeneratedAttribute
public final function set WorkstationId (value : String)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.XmlStoredProcedureRootNodeName Property

Specifies the XML root node name of the stored procedures for the database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
Public Property XmlStoredProcedureRootNodeName As String
```

C#

```
public string XmlStoredProcedureRootNodeName { get; set; }
```

C++

```
public:
virtual property String^ XmlStoredProcedureRootNodeName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_XmlStoredProcedureRootNodeName ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_XmlStoredProcedureRootNodeName (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get XmlStoredProcedureRootNodeName () : String

CompilerGeneratedAttribute
public final function set XmlStoredProcedureRootNodeName (value : String)
```

Property Value

Returns a [String](#) containing the root node name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of

Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElement.XmlStoredProcedureRootNodeNamespace Property

Specifies the XML stored procedure root node namespace.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
Public Property XmlStoredProcedureRootNodeNamespace As String
```

C#

```
public string XmlStoredProcedureRootNodeNamespace { get; set; }
```

C++

```
public:
virtual property String^ XmlStoredProcedureRootNodeNamespace {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final String get_XmlStoredProcedureRootNodeNamespace ()

/** @property */
/** @attribute CompilerGeneratedAttribute() */
public final void set_XmlStoredProcedureRootNodeNamespace (String value)
```

JScript

```
CompilerGeneratedAttribute
public final function get XmlStoredProcedureRootNodeNamespace () : String

CompilerGeneratedAttribute
public final function set XmlStoredProcedureRootNodeNamespace (value : String)
```

Property Value

Returns a [String](#) containing the namespace name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with

SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[SqlAdapterBindingElement Class](#)

[SqlAdapterBindingElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement Class

Represents the SQL adapter binding.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class SqlAdapterBindingElementExtensionElement
    Inherits BindingElementExtensionElement
```

C#

```
public sealed class SqlAdapterBindingElementExtensionElement : BindingElementExtensionElement
```

C++

```
public ref class SqlAdapterBindingElementExtensionElement sealed : public BindingElementExtensionElement
```

J#

```
public final class SqlAdapterBindingElementExtensionElement extends BindingElementExtensionElement
```

JScript

```
public final class SqlAdapterBindingElementExtensionElement extends BindingElementExtensionElement
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BindingElementExtensionElement](#)

Microsoft.Adapters.Sql.SqlAdapterBindingElementExtensionElement

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement Members

Represents the SQL adapter binding.

The following tables list the members exposed by the [SqlAdapterBindingElementExtensionElement](#) type.

Public Constructors

Name	Description
SqlAdapterBindingElementExtensionElement	Creates a new instance of the SqlAdapterBindingElementExtensionElement class, using the default parameters.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 BindingElementType	Overridden. Specifies the binding element type.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

Top

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from ConfigurationElement)

[Top](#)

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	Overridden. Applies the configuration to the specified binding element.
	CopyFrom	Overridden. copies the specified extension element to the current instance.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

[Top](#)

Protected Methods

	Name	Description
	CreateBindingElement	(inherited from BindingElementExtensionElement)
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from BindingElementExtensionElement)
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

[Top](#)

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement Constructor

Creates a new instance of the `SqlAdapterBindingElementExtensionElement` class, using the default parameters.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterBindingElementExtensionElement ()
```

C++

```
public:  
SqlAdapterBindingElementExtensionElement ()
```

J#

```
public SqlAdapterBindingElementExtensionElement ()
```

JScript

```
public function SqlAdapterBindingElementExtensionElement ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	ApplyConfiguration	Overridden. Applies the configuration to the specified binding element.
	CopyFrom	Overridden. copies the specified extension element to the current instance.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBindingElement	(inherited from BindingElementExtensionElement)
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	InitializeFrom	(inherited from BindingElementExtensionElement)
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.ApplyConfiguration Method

Applies the configuration to the specified binding element.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub ApplyConfiguration ( _  
    bindingElement As BindingElement _  
)
```

C#

```
public override void ApplyConfiguration (  
    BindingElement bindingElement  
)
```

C++

```
public:  
virtual void ApplyConfiguration (  
    BindingElement^ bindingElement  
) override
```

J#

```
public void ApplyConfiguration (  
    BindingElement bindingElement  
)
```

JScript

```
public override function ApplyConfiguration (  
    bindingElement : BindingElement  
)
```

Parameters

bindingElement

The binding element to apply the configuration to.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows

Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008
See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.CopyFrom Method

copies the specified extension element to the current instance.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

the extension element to copy.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement Properties

Public Properties (see also [Protected Properties](#))

Name	Description
 AcceptCredentialsInUri	
 AllowIdentityInsert	Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.
 BatchSize	Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.
 BindingElementType	Overridden. Specifies the binding element type.
 ChunkSize	Specifies the buffer size used for Set<column_name> operations.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 EnableBizTalkCompatibilityMode	Gets or sets a value indicating if the BizTalk compatibility mode has been set.
 EnablePerformanceCounters	Gets or sets a value specifying whether to enable the WCF LOB Adapter SDK performance counters and the SQL adapter LOB latency performance counter.
 Encrypt	Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.
 InboundOperationType	Specifies if you want to perform a polling, typed polling, or notification operation.
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 MaxConnectionPoolSize	Specifies the maximum number of connections allowed in a connection pool for a specific connection string.
 NotificationStatement	Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.
 NotifyOnListenerStart	Specifies whether the adapter sends a notification message when the listener is started.
 PolledDataAvailableStatement	Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.
 PollingIntervalInSeconds	Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.
 PollingStatement	Specifies the SQL statement to poll a SQL Server database table.
 PollWhileDataFound	Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.
 UseAmbientTransaction	Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.
 UseDatabaseNameInXsdNamespace	Specifies whether the XSD generated for a particular artifact contains the database name.
 WorkstationId	
 XmlStoredProcedureRootNodeName	Specifies the XML root node name of the stored procedures for the database.
 XmlStoredProcedureRootNodeNamespace	Specifies the XML stored procedure root node namespace.

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from ConfigurationElement)

See Also

Reference

SqlAdapterBindingElementExtensionElement Class
Microsoft.Adapters.Sql Namespace

SqlAdapterBindingElementExtensionElement.AcceptCredentialsInUri Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.AllowIdentityInsert Property

Gets or sets a value indicating whether you can insert values for identity columns during Insert and update operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Miscellaneous")> _
<ConfigurationPropertyAttribute("allowIdentityInsert", DefaultValue:=False)> _
Public Property AllowIdentityInsert As Boolean
```

C#

```
[CategoryAttribute("Miscellaneous")]
[ConfigurationPropertyAttribute("allowIdentityInsert", DefaultValue=false)]
public bool AllowIdentityInsert { get; set; }
```

C++

```
[CategoryAttribute(L"Miscellaneous")]
[ConfigurationPropertyAttribute(L"allowIdentityInsert", DefaultValue=false)]
public:
virtual property bool AllowIdentityInsert {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_AllowIdentityInsert ()

/** @property */
public final void set_AllowIdentityInsert (boolean value)
```

JScript

```
public final function get AllowIdentityInsert () : boolean

public final function set AllowIdentityInsert (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if you can insert values for identity columns. The default is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows

Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.BatchSize Property

Specifies the batch size for multiple record Insert and delete operations on a table or view in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("batchSize", DefaultValue:=20)> _
<CategoryAttribute("Buffering")> _
Public Property BatchSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("batchSize", DefaultValue=20)]
[CategoryAttribute("Buffering")]
public int BatchSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"batchSize", DefaultValue=20)]
[CategoryAttribute(L"Buffering")]
public:
virtual property int BatchSize {
int get () sealed;
void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_BatchSize ()

/** @property */
public final void set_BatchSize (int value)
```

JScript

```
public final function get BatchSize () : int

public final function set BatchSize (value : int)
```

Property Value

Returns [Int32](#).

Remarks

The default is two.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.BindingElementType Property

Specifies the binding element type.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Overrides ReadOnly Property BindingElementType As Type
```

C#

```
public override Type BindingElementType { get; }
```

C++

```
public:  
virtual property Type^ BindingElementType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_BindingElementType ()
```

JScript

```
public override function get BindingElementType () : Type
```

Property Value

Returns a [Type](#) containing the binding element.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

SqlAdapterBindingElementExtensionElement.ChunkSize Property

Specifies the buffer size used for Set<column_name> operations.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("chunkSize", DefaultValue:=4194304)> _
<CategoryAttribute("Buffering")> _
Public Property ChunkSize As Integer
```

C#

```
[ConfigurationPropertyAttribute("chunkSize", DefaultValue=4194304)]
[CategoryAttribute("Buffering")]
public int ChunkSize { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"chunkSize", DefaultValue=4194304)]
[CategoryAttribute(L"Buffering")]
public:
virtual property int ChunkSize {
int get () sealed;
void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_ChunkSize ()

/** @property */
public final void set_ChunkSize (int value)
```

JScript

```
public final function get ChunkSize () : int

public final function set ChunkSize (value : int)
```

Property Value

Returns a [Int32](#) containing the buffer size.

Remarks

The default is 419304 bytes. A higher value may improve performance, but affects memory consumption.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.EnableBizTalkCompatibilityMode Property

Gets or sets a value indicating if the BizTalk compatibility mode has been set.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue:=True)> _
<CategoryAttribute("BizTalk")> _
Public Property EnableBizTalkCompatibilityMode As Boolean
```

C#

```
[ConfigurationPropertyAttribute("enableBizTalkCompatibilityMode", DefaultValue=true)]
[CategoryAttribute("BizTalk")]
public bool EnableBizTalkCompatibilityMode { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"enableBizTalkCompatibilityMode", DefaultValue=true)]
[CategoryAttribute(L"BizTalk")]
public:
virtual property bool EnableBizTalkCompatibilityMode {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnableBizTalkCompatibilityMode ()

/** @property */
public final void set_EnableBizTalkCompatibilityMode (boolean value)
```

JScript

```
public final function get EnableBizTalkCompatibilityMode () : boolean

public final function set EnableBizTalkCompatibilityMode (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the compatibility mode has been set.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.EnablePerformanceCounters Property

Gets or sets a value specifying whether to enable the WCF LOB Adpater SDK performance counters and the SQL adapter LOB latency performance counter.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<CategoryAttribute("Diagnostics")> _
<ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue:=False)> _
Public Property EnablePerformanceCounters As Boolean
```

C#

```
[CategoryAttribute("Diagnostics")]
[ConfigurationPropertyAttribute("enablePerformanceCounters", DefaultValue=false)]
public bool EnablePerformanceCounters { get; set; }
```

C++

```
[CategoryAttribute(L"Diagnostics")]
[ConfigurationPropertyAttribute(L"enablePerformanceCounters", DefaultValue=false)]
public:
virtual property bool EnablePerformanceCounters {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_EnablePerformanceCounters ()

/** @property */
public final void set_EnablePerformanceCounters (boolean value)
```

JScript

```
public final function get EnablePerformanceCounters () : boolean

public final function set EnablePerformanceCounters (value : boolean)
```

Property Value

Returns a [Boolean](#), indicating if the performance counters have been enabled.

Remarks

The default value is false; performance counters are disabled. The LOB latency performance counter measures the total time spent by the SQL adapter in making calls to the SQL server database.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.Encrypt Property

Specifies whether SQL Server (with a valid certificate installed) uses SSL encryption for all data transfers between SQL Server and the client.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("encrypt", DefaultValue:=False)> _
<CategoryAttribute("Connection")> _
Public Property Encrypt As Boolean
```

C#

```
[ConfigurationPropertyAttribute("encrypt", DefaultValue=false)]
[CategoryAttribute("Connection")]
public bool Encrypt { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"encrypt", DefaultValue=false)]
[CategoryAttribute(L"Connection")]
public:
virtual property bool Encrypt {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_Encrypt ()

/** @property */
public final void set_Encrypt (boolean value)
```

JScript

```
public final function get Encrypt () : boolean

public final function set Encrypt (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if SQL server uses SSL encryption; the default value is false.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.InboundOperationType Property

Specifies if you want to perform a polling, typed polling, or notification operation.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("inboundOperationType", DefaultValue:=InboundOperation.Polling)> _
<CategoryAttribute("Inbound")> _
Public Property InboundOperationType As InboundOperation
```

C#

```
[ConfigurationPropertyAttribute("inboundOperationType", DefaultValue=InboundOperation.Polling)]
[CategoryAttribute("Inbound")]
public InboundOperation InboundOperationType { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"inboundOperationType", DefaultValue=InboundOperation::Polling)]
[CategoryAttribute(L"Inbound")]
public:
virtual property InboundOperation InboundOperationType {
    InboundOperation get () sealed;
    void set (InboundOperation value) sealed;
}
```

J#

```
/** @property */
public final InboundOperation get_InboundOperationType ()

/** @property */
public final void set_InboundOperationType (InboundOperation value)
```

JScript

```
public final function get InboundOperationType () : InboundOperation
public final function set InboundOperationType (value : InboundOperation)
```

Property Value

Returns a [InboundOperation](#) containing the operation type; the default is Polling.

Thread Safety/Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.MaxConnectionPoolSize Property

Specifies the maximum number of connections allowed in a connection pool for a specific connection string.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("Connection")> _
<ConfigurationPropertyAttribute("maxConnectionPoolSize", DefaultValue:=100)> _
Public Property MaxConnectionPoolSize As Integer
```

C#

```
[CategoryAttribute("Connection")]
[ConfigurationPropertyAttribute("maxConnectionPoolSize", DefaultValue=100)]
public int MaxConnectionPoolSize { get; set; }
```

C++

```
[CategoryAttribute(L"Connection")]
[ConfigurationPropertyAttribute(L"maxConnectionPoolSize", DefaultValue=100)]
public:
virtual property int MaxConnectionPoolSize {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_MaxConnectionPoolSize ()

/** @property */
public final void set_MaxConnectionPoolSize (int value)
```

JScript

```
public final function get MaxConnectionPoolSize () : int
public final function set MaxConnectionPoolSize (value : int)
```

Property Value

Returns a [Int32](#) specifying the maximum number of connections; the default is 100.

Remarks

This property is used for performance tuning.

Use this property judiciously. If you set the value too large, you may exhaust the number of connections available.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.NotificationStatement Property

Specifies the SQL statement (SELECT or EXEC <stored procedure>) used to register for SQL Server notifications.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Notification (Inbound)")> _
<ConfigurationPropertyAttribute("notificationStatement", DefaultValue:=<string>)> _
Public Property NotificationStatement As String
```

C#

```
[CategoryAttribute("Notification (Inbound)")]
[ConfigurationPropertyAttribute("notificationStatement", DefaultValue="")]
public string NotificationStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Notification (Inbound)")]
[ConfigurationPropertyAttribute(L"notificationStatement", DefaultValue=L"")]
public:
virtual property String^ NotificationStatement {
    String^ get () sealed;
    void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_NotificationStatement ()

/** @property */
public final void set_NotificationStatement (String value)
```

JScript

```
public final function get NotificationStatement () : String

public final function set NotificationStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.NotifyOnListenerStart Property

Specifies whether the adapter sends a notification message when the listener is started.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue:=True)> _
<CategoryAttribute("Notification (Inbound)")> _
Public Property NotifyOnListenerStart As Boolean
```

C#

```
[ConfigurationPropertyAttribute("notifyOnListenerStart", DefaultValue=true)]
[CategoryAttribute("Notification (Inbound)")]
public bool NotifyOnListenerStart { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"notifyOnListenerStart", DefaultValue=true)]
[CategoryAttribute(L"Notification (Inbound)")]
public:
virtual property bool NotifyOnListenerStart {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_NotifyOnListenerStart ()

/** @property */
public final void set_NotifyOnListenerStart (boolean value)
```

JScript

```
public final function get NotifyOnListenerStart () : boolean

public final function set NotifyOnListenerStart (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating if the adapter sends the notification message.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.PolledDataAvailableStatement Property

Specifies the SQL statement executed to determine whether any data is available for polling for a specific table in a SQL Server database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("Polling (Inbound)")> _
<ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue="")> _
Public Property PolledDataAvailableStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[ConfigurationPropertyAttribute("polledDataAvailableStatement", DefaultValue="")]
public string PolledDataAvailableStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[ConfigurationPropertyAttribute(L"polledDataAvailableStatement", DefaultValue=L"")]
public:
virtual property String^ PolledDataAvailableStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PolledDataAvailableStatement ()

/** @property */
public final void set_PolledDataAvailableStatement (String value)
```

JScript

```
public final function get PolledDataAvailableStatement () : String
public final function set PolledDataAvailableStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

The specified statement must return a result set consisting of rows and columns. The value in the first cell of the result set indicates whether the adapter executes the SQL statement specified for the [PollingStatement](#) binding property. If the first cell of the result contains a positive value, the adapter executed the polling statement.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2

(except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.PollingIntervalInSeconds Property

Specifies the interval, in seconds, at which the SQL adapter executes the statement specified for the PolledDataAvailableStatement binding property.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<CategoryAttribute("Polling (Inbound)")> _
<ConfigurationPropertyAttribute("pollingIntervalInSeconds", DefaultValue:=30)> _
Public Property PollingIntervalInSeconds As Integer
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[ConfigurationPropertyAttribute("pollingIntervalInSeconds", DefaultValue=30)]
public int PollingIntervalInSeconds { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[ConfigurationPropertyAttribute(L"pollingIntervalInSeconds", DefaultValue=30)]
public:
virtual property int PollingIntervalInSeconds {
    int get () sealed;
    void set (int value) sealed;
}
```

J#

```
/** @property */
public final int get_PollingIntervalInSeconds ()

/** @property */
public final void set_PollingIntervalInSeconds (int value)
```

JScript

```
public final function get PollingIntervalInSeconds () : int
public final function set PollingIntervalInSeconds (value : int)
```

Property Value

Returns a [Int32](#) containing the interval.

Remarks

The default is 30 seconds. The polling interval determines the time interval between successive polls. If the statement is executed within the specified interval, the adapter is inactive for the remaining time in the interval.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003

with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.PollingStatement Property

Specifies the SQL statement to poll a SQL Server database table.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<CategoryAttribute("Polling (Inbound)")> _
<ConfigurationPropertyAttribute("pollingStatement", DefaultValue:= "")> _
Public Property PollingStatement As String
```

C#

```
[CategoryAttribute("Polling (Inbound)")]
[ConfigurationPropertyAttribute("pollingStatement", DefaultValue="")]
public string PollingStatement { get; set; }
```

C++

```
[CategoryAttribute(L"Polling (Inbound)")]
[ConfigurationPropertyAttribute(L"pollingStatement", DefaultValue=L"")]
public:
virtual property String^ PollingStatement {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_PollingStatement ()

/** @property */
public final void set_PollingStatement (String value)
```

JScript

```
public final function get PollingStatement () : String
public final function set PollingStatement (value : String)
```

Property Value

Returns a [String](#) containing the SQL statement.

Remarks

You can specify a simple SELECT statement or stored procedures for the polling statement. The default is null. You must specify a value for PollingStatement to enable polling. The polling statement is executed only if there is data available for polling,

which is determined by the PolledDataAvailableStatement binding property.

You can specify any number of SQL statements separated by a semicolon. You can use the polling statement to read or update data in a SQL Server database table. The SQL adapter executes the polling statements inside one transaction. When the adapter is used with BizTalk Server, the same transaction is used to submit messages from SQL Server to the BizTalk message box.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.PollWhileDataFound Property

Specifies whether the SQL adapter ignores the polling interval and continuously executes the SQL statement specified for the PolledDataAvailableStatement binding property, if data is available in the table being polled.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue:=False)> _
<CategoryAttribute("Polling (Inbound)")> _
Public Property PollWhileDataFound As Boolean
```

C#

```
[ConfigurationPropertyAttribute("pollWhileDataFound", DefaultValue=false)]
[CategoryAttribute("Polling (Inbound)")]
public bool PollWhileDataFound { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"pollWhileDataFound", DefaultValue=false)]
[CategoryAttribute(L"Polling (Inbound)")]
public:
virtual property bool PollWhileDataFound {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_PollWhileDataFound ()

/** @property */
public final void set_PollWhileDataFound (boolean value)
```

JScript

```
public final function get PollWhileDataFound () : boolean

public final function set PollWhileDataFound (value : boolean)
```

Property Value

Returns a [Boolean](#) specifying whether to ignore the polling interval; the default is false.

Remarks

If no data is available in the table, the adapter reverts to execute the SQL statement at the specified polling interval.

Consider a scenario where the polling interval is set to 60 seconds, and the statement specified for PolledDataAvailableStatement returns that data is available for polling. The adapter then executes the statement specified for the PollingStatement binding

property. Assuming that the adapter takes just 10 seconds to execute the polling statement, it will now have to wait for 50 seconds before executing the PolledDataAvailableStatement again, and then subsequently execute the polling statement. Instead, to optimize the performance you can set the PollWhileDataFound binding property to true so that the adapter can start executing the next polling cycle as soon as the previous polling cycle ends.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.UseAmbientTransaction Property

Specifies whether the SQL adapter performs the operations using the transactional context provided by the caller.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue:=True)> _
<CategoryAttribute("Transaction")> _
Public Property UseAmbientTransaction As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useAmbientTransaction", DefaultValue=true)]
[CategoryAttribute("Transaction")]
public bool UseAmbientTransaction { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useAmbientTransaction", DefaultValue=true)]
[CategoryAttribute(L"Transaction")]
public:
virtual property bool UseAmbientTransaction {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseAmbientTransaction ()

/** @property */
public final void set_UseAmbientTransaction (boolean value)
```

JScript

```
public final function get UseAmbientTransaction () : boolean

public final function set UseAmbientTransaction (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the SQL adapter performs operations using a transactional context; the default is true.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.UseDatabaseNameInXsdNamespace Property

Specifies whether the XSD generated for a particular artifact contains the database name.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("useDatabaseNameInXsdNamespace", DefaultValue:=False)> _
<CategoryAttribute("Metadata")> _
Public Property UseDatabaseNameInXsdNamespace As Boolean
```

C#

```
[ConfigurationPropertyAttribute("useDatabaseNameInXsdNamespace", DefaultValue=false)]
[CategoryAttribute("Metadata")]
public bool UseDatabaseNameInXsdNamespace { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"useDatabaseNameInXsdNamespace", DefaultValue=false)]
[CategoryAttribute(L"Metadata")]
public:
virtual property bool UseDatabaseNameInXsdNamespace {
    bool get () sealed;
    void set (bool value) sealed;
}
```

J#

```
/** @property */
public final boolean get_UseDatabaseNameInXsdNamespace ()

/** @property */
public final void set_UseDatabaseNameInXsdNamespace (boolean value)
```

JScript

```
public final function get UseDatabaseNameInXsdNamespace () : boolean

public final function set UseDatabaseNameInXsdNamespace (value : boolean)
```

Property Value

Returns a [Boolean](#) indicating whether the XSD contains the database name; the default is false.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.WorkstationId Property

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("workstationId", DefaultValue:= "") > _
<CategoryAttribute("Connection") > _
Public Property WorkstationId As String
```

C#

```
[ConfigurationPropertyAttribute("workstationId", DefaultValue="")]
[CategoryAttribute("Connection")]
public string WorkstationId { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"workstationId", DefaultValue=L"")]
[CategoryAttribute(L"Connection")]
public:
virtual property String^ WorkstationId {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_WorkstationId ()

/** @property */
public final void set_WorkstationId (String value)
```

JScript

```
public final function get WorkstationId () : String
public final function set WorkstationId (value : String)
```

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.XmlStoredProcedureRootNodeName Property

Specifies the XML root node name of the stored procedures for the database.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeName", DefaultValue:= "") > _
<CategoryAttribute("FOR XML")> _
Public Property XmlStoredProcedureRootNodeName As String
```

C#

```
[ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeName", DefaultValue="")]
[CategoryAttribute("FOR XML")]
public string XmlStoredProcedureRootNodeName { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"xmlStoredProcedureRootNodeName", DefaultValue=L"")]
[CategoryAttribute(L"FOR XML")]
public:
virtual property String^ XmlStoredProcedureRootNodeName {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_XmlStoredProcedureRootNodeName ()

/** @property */
public final void set_XmlStoredProcedureRootNodeName (String value)
```

JScript

```
public final function get XmlStoredProcedureRootNodeName () : String
public final function set XmlStoredProcedureRootNodeName (value : String)
```

Property Value

Returns a [String](#) containing the root node name.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterBindingElementExtensionElement.XmlStoredProcedureRootNodeNamespace Property

Specifies the XML stored procedure root node namespace.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
<ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeNamespace", DefaultValue="")> _
<CategoryAttribute("FOR XML")> _
Public Property XmlStoredProcedureRootNodeNamespace As String
```

C#

```
[ConfigurationPropertyAttribute("xmlStoredProcedureRootNodeNamespace", DefaultValue="")]
[CategoryAttribute("FOR XML")]
public string XmlStoredProcedureRootNodeNamespace { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"xmlStoredProcedureRootNodeNamespace", DefaultValue=L"")]
[CategoryAttribute(L"FOR XML")]
public:
virtual property String^ XmlStoredProcedureRootNodeNamespace {
String^ get () sealed;
void set (String^ value) sealed;
}
```

J#

```
/** @property */
public final String get_XmlStoredProcedureRootNodeNamespace ()

/** @property */
public final void set_XmlStoredProcedureRootNodeNamespace (String value)
```

JScript

```
public final function get XmlStoredProcedureRootNodeNamespace () : String
public final function set XmlStoredProcedureRootNodeNamespace (value : String)
```

Property Value

Returns a [String](#) containing the namespace name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition);

The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions;

The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition);

The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions;

The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterBindingElementExtensionElement Class](#)

[SqlAdapterBindingElementExtensionElement Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri Class

Contains the SQL adapter connection URI.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public NotInheritable Class SqlAdapterConnectionUri  
    Inherits ConnectionUri
```

C#

```
public sealed class SqlAdapterConnectionUri : ConnectionUri
```

C++

```
public ref class SqlAdapterConnectionUri sealed : public ConnectionUri
```

J#

```
public final class SqlAdapterConnectionUri extends ConnectionUri
```

JScript

```
public final class SqlAdapterConnectionUri extends ConnectionUri
```

Inheritance Hierarchy [System.Object](#)

[Microsoft.ServiceModel.Channels.Common.ConnectionUri](#)

Microsoft.Adapters.Sql.SqlAdapterConnectionUri

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri Members

Contains the SQL adapter connection URI.

The following tables list the members exposed by the [SqlAdapterConnectionUri](#) type.

Public Constructors

Name	Description
SqlAdapterConnectionUri	Contains a new instance of the SqlAdapterConnectionUri class.

[Top](#)

Public Properties

Name	Description
 FailoverPartner	Gets or sets the name of the failover SQL Server database to connect to if the primary SQL Server database is not available.
 InboundId	Gets or sets the identifier you add to the connection URI to make it unique.
 InitialCatalog	Gets or sets the initial catalog.
 InstanceName	Gets or sets the name of the SQL Server instance to connect to.
 SampleUriString	Overridden. Gets a sample URI string.
 SecureUriString	(inherited from ConnectionUri)
 Server	Gets or sets the name of the server on which SQL Server is installed.
 Uri	Overridden. Gets or sets the complete connection URI.

[Top](#)

Public Methods (see also [Protected Methods](#))

Name	Description
 Equals	Overloaded. (inherited from Object)
 GetAttributes	(inherited from ConnectionUri)
 GetClassName	(inherited from ConnectionUri)
 GetComponentName	(inherited from ConnectionUri)
 GetConverter	(inherited from ConnectionUri)
 GetDefaultEvent	(inherited from ConnectionUri)
 GetDefaultProperty	(inherited from ConnectionUri)
 GetEditor	(inherited from ConnectionUri)
 GetEvents	Overloaded. (inherited from ConnectionUri)
 GetHashCode	(inherited from Object)
 GetProperties	Overloaded. (inherited from ConnectionUri)
 GetPropertyOwner	(inherited from ConnectionUri)
 GetType	(inherited from Object)
  ReferenceEquals	(inherited from Object)
 ToString	(inherited from ConnectionUri)

[Top](#)

Protected Methods

Name	Description
 Finalize	(inherited from Object)
  GetQueryStringValue	(inherited from ConnectionUri)
  GetUserInfoValue	(inherited from ConnectionUri)
 MemberwiseClone	(inherited from Object)

[Top](#)

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri Constructor

Contains a new instance of the SqlAdapterConnectionUri class.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterConnectionUri ()
```

C++

```
public:  
SqlAdapterConnectionUri ()
```

J#

```
public SqlAdapterConnectionUri ()
```

JScript

```
public function SqlAdapterConnectionUri ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	Equals	Overloaded. (inherited from Object)
	GetAttributes	(inherited from ConnectionUri)
	GetClassName	(inherited from ConnectionUri)
	GetComponentName	(inherited from ConnectionUri)
	GetConverter	(inherited from ConnectionUri)
	GetDefaultEvent	(inherited from ConnectionUri)
	GetDefaultProperty	(inherited from ConnectionUri)
	GetEditor	(inherited from ConnectionUri)
	GetEvents	Overloaded. (inherited from ConnectionUri)
	GetHashCode	(inherited from Object)
	GetProperties	Overloaded. (inherited from ConnectionUri)
	GetPropertyOwner	(inherited from ConnectionUri)
	GetType	(inherited from Object)
	 ReferenceEquals	(inherited from Object)
	ToString	(inherited from ConnectionUri)

Protected Methods

	Name	Description
	Finalize	(inherited from Object)
	 GetQueryStringValue	(inherited from ConnectionUri)
	 GetUserInfoValue	(inherited from ConnectionUri)
	MemberwiseClone	(inherited from Object)

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri Properties

Public Properties

	Name	Description
	FailoverPartner	Gets or sets the name of the failover SQL Server database to connect to if the primary SQL Server database is not available.
	InboundId	Gets or sets the identifier you add to the connection URI to make it unique.
	InitialCatalog	Gets or sets the initial catalog.
	InstanceName	Gets or sets the name of the SQL Server instance to connect to.
	SampleUriString	Overridden. Gets a sample URI string.
	SecureUriString	(inherited from ConnectionUri)
	Server	Gets or sets the name of the server on which SQL Server is installed.
	Uri	Overridden. Gets or sets the complete connection URI.

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.FailoverPartner Property

Gets or sets the name of the failover SQL Server database to connect to if the primary SQL Server database is not available.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property FailoverPartner As String
```

C#

```
public string FailoverPartner { get; set; }
```

C++

```
public:  
property String^ FailoverPartner {  
    String^ get ();  
    void set (String^ value);  
}
```

J#

```
/** @property */  
public String get_FailoverPartner ()  
  
/** @property */  
public void set_FailoverPartner (String value)
```

JScript

```
public function get FailoverPartner () : String  
  
public function set FailoverPartner (value : String)
```

Property Value

Returns a [String](#) containing the database name.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.InboundId Property

Gets or sets the identifier you add to the connection URI to make it unique.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property InboundId As String
```

C#

```
public string InboundId { get; set; }
```

C++

```
public:  
property String^ InboundId {  
    String^ get ();  
    void set (String^ value);  
}
```

J#

```
/** @property */  
public String get_InboundId ()  
  
/** @property */  
public void set_InboundId (String value)
```

JScript

```
public function get InboundId () : String  
  
public function set InboundId (value : String)
```

Property Value

Returns a [String](#) containing the inbound ID.

Remarks

You must provide this connection parameter if you want to generate metadata for the TypedPolling inbound operation. Also, in a BizTalk application, if you have multiple receive locations polling the same database, the inbound ID makes the connection URI unique, thereby enabling adapter clients to receive polling messages from the same database on different receive locations.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.InitialCatalog Property

Gets or sets the initial catalog.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property InitialCatalog As String
```

C#

```
public string InitialCatalog { get; set; }
```

C++

```
public:  
property String^ InitialCatalog {  
    String^ get ();  
    void set (String^ value);  
}
```

J#

```
/** @property */  
public String get_InitialCatalog ()  
  
/** @property */  
public void set_InitialCatalog (String value)
```

JScript

```
public function get InitialCatalog () : String  
  
public function set InitialCatalog (value : String)
```

Property Value

Returns a [String](#) containing the name of the initial catalog.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.InstanceName Property

Gets or sets the name of the SQL Server instance to connect to.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property InstanceName As String
```

C#

```
public string InstanceName { get; set; }
```

C++

```
public:  
property String^ InstanceName {  
    String^ get ();  
    void set (String^ value);  
}
```

J#

```
/** @property */  
public String get_InstanceName ()  
  
/** @property */  
public void set_InstanceName (String value)
```

JScript

```
public function get InstanceName () : String  
  
public function set InstanceName (value : String)
```

Property Value

Returns a [String](#) containing the server instance name.

Remarks

If you do not specify a value, the adapter connects to the default database instance.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.SampleUriString Property

Gets a sample URI string.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides NotOverridable ReadOnly Property SampleUriString As String
```

C#

```
public override sealed string SampleUriString { get; }
```

C++

```
public:  
virtual property String^ SampleUriString {  
    String^ get () override sealed;  
}
```

J#

```
/** @property */  
public final String get_SampleUriString ()
```

JScript

```
public override final function get SampleUriString () : String
```

Property Value

Returns a [String](#) containing a sample URI string.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.Server Property

Gets or sets the name of the server on which SQL Server is installed.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Property Server As String
```

C#

```
public string Server { get; set; }
```

C++

```
public:  
property String^ Server {  
    String^ get ();  
    void set (String^ value);  
}
```

J#

```
/** @property */  
public String get_Server ()  
  
/** @property */  
public void set_Server (String value)
```

JScript

```
public function get Server () : String  
  
public function set Server (value : String)
```

Property Value

Returns a [String](#) containing the name of the server.

Remarks

If you do not specify a value, the adapter assumes the server name as "localhost" and establishes a connection with the SQL Server database on the local server.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterConnectionUri.Uri Property

Gets or sets the complete connection URI.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public Overrides NotOverridable Property Uri As Uri
```

C#

```
public override sealed Uri Uri { get; set; }
```

C++

```
public:
virtual property Uri^ Uri {
    Uri^ get () override sealed;
    void set (Uri^ value) override sealed;
}
```

J#

```
/** @property */
public final Uri get Uri ()

/** @property */
public final void set Uri (Uri value)
```

JScript

```
public override final function get Uri () : Uri

public override final function set Uri (value : Uri)
```

Property Value

Returns a [Uri](#) containing the URI.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of

Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterConnectionUri Class](#)

[SqlAdapterConnectionUri Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior Class

Describes the SQL adapter inbound transaction behavior.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

Public NotInheritable Class SqlAdapterInboundTransactionBehavior
    Inherits BehaviorExtensionElement
    Implements IServiceBehavior
```

C#

```
public sealed class SqlAdapterInboundTransactionBehavior : BehaviorExtensionElement, IServiceBehavior
```

C++

```
public ref class SqlAdapterInboundTransactionBehavior sealed : public BehaviorExtensionElement, IServiceBehavior
```

J#

```
public final class SqlAdapterInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

JScript

```
public final class SqlAdapterInboundTransactionBehavior extends BehaviorExtensionElement implements IServiceBehavior
```

Inheritance Hierarchy [System.Object](#)

[System.Configuration.ConfigurationElement](#)

[System.ServiceModel.Configuration.ServiceModelExtensionElement](#)

[System.ServiceModel.Configuration.BehaviorExtensionElement](#)

Microsoft.Adapters.Sql.SqlAdapterInboundTransactionBehavior

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-

bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior Members

Describes the SQL adapter inbound transaction behavior.

The following tables list the members exposed by the [SqlAdapterInboundTransactionBehavior](#) type.

Public Constructors

Name	Description
SqlAdapterInboundTransactionBehavior	Creates a new instance of the SqlAdapterInboundTransactionBehavior class.

Top

Public Properties (see also [Protected Properties](#))

Name	Description
 BehaviorType	Overridden. Gets the type of behavior for the inbound transaction.
 ConfigurationElementName	(inherited from ServiceModelExtensionElement)
 ElementInformation	(inherited from ConfigurationElement)
 LockAllAttributesExcept	(inherited from ConfigurationElement)
 LockAllElementsExcept	(inherited from ConfigurationElement)
 LockAttributes	(inherited from ConfigurationElement)
 LockElements	(inherited from ConfigurationElement)
 LockItem	(inherited from ConfigurationElement)
 TransactionIsolationLevel	Gets or sets the transaction isolation level.
 TransactionTimeout	Gets or sets the transaction timeout level.

Top

Protected Properties

Name	Description
 ElementProperty	(inherited from ConfigurationElement)
 EvaluationContext	(inherited from ConfigurationElement)
 Item	Overloaded. (inherited from ConfigurationElement)
 Properties	(inherited from ConfigurationElement)

Top

Public Methods (see also [Protected Methods](#))

Name	Description
 CopyFrom	Overridden. Copies the transaction behavior from the specified extension element to the current instance.
 Equals	Overloaded. (inherited from ConfigurationElement)
 GetHashCode	(inherited from ConfigurationElement)
 GetType	(inherited from Object)
 IsReadOnly	(inherited from ConfigurationElement)
 ReferenceEquals	(inherited from Object)
 ToString	(inherited from Object)

Top

Protected Methods

Name	Description
 CreateBehavior	(inherited from BehaviorExtensionElement)
 DeserializeElement	(inherited from ConfigurationElement)
 Finalize	(inherited from Object)
 Init	(inherited from ConfigurationElement)
 InitializeDefault	(inherited from ConfigurationElement)
 IsModified	(inherited from ServiceModelExtensionElement)
 ListErrors	(inherited from ConfigurationElement)
 MemberwiseClone	(inherited from Object)
 OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
 OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
 OnRequiredPropertyNotFound	(inherited from ConfigurationElement)

 PostDeserialize	(inherited from ConfigurationElement)
 PreSerialize	(inherited from ConfigurationElement)
 Reset	(inherited from ServiceModelExtensionElement)
 ResetModified	(inherited from ConfigurationElement)
 SerializeElement	(inherited from ServiceModelExtensionElement)
 SerializeToXmlElement	(inherited from ConfigurationElement)
 SetPropertyValue	(inherited from ConfigurationElement)
 SetReadOnly	(inherited from ConfigurationElement)
 Unmerge	(inherited from ConfigurationElement)

[Top](#)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

[Top](#)

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior Constructor

Creates a new instance of the SqlAdapterInboundTransactionBehavior class.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
Public Sub New
```

C#

```
public SqlAdapterInboundTransactionBehavior ()
```

C++

```
public:  
SqlAdapterInboundTransactionBehavior ()
```

J#

```
public SqlAdapterInboundTransactionBehavior ()
```

JScript

```
public function SqlAdapterInboundTransactionBehavior ()
```

RemarksThe default constructor initializes any fields to their default values.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior Methods

Public Methods (see also [Protected Methods](#))

	Name	Description
	CopyFrom	Overridden. Copies the transaction behavior from the specified extension element to the current instance.
	Equals	Overloaded. (inherited from ConfigurationElement)
	GetHashCode	(inherited from ConfigurationElement)
	GetType	(inherited from Object)
	IsReadOnly	(inherited from ConfigurationElement)
 	ReferenceEquals	(inherited from Object)
	ToString	(inherited from Object)

Protected Methods

	Name	Description
	CreateBehavior	(inherited from BehaviorExtensionElement)
	DeserializeElement	(inherited from ConfigurationElement)
	Finalize	(inherited from Object)
	Init	(inherited from ConfigurationElement)
	InitializeDefault	(inherited from ConfigurationElement)
	IsModified	(inherited from ServiceModelExtensionElement)
	ListErrors	(inherited from ConfigurationElement)
	MemberwiseClone	(inherited from Object)
	OnDeserializeUnrecognizedAttribute	(inherited from ConfigurationElement)
	OnDeserializeUnrecognizedElement	(inherited from ConfigurationElement)
	OnRequiredPropertyNotFound	(inherited from ConfigurationElement)
	PostDeserialize	(inherited from ConfigurationElement)
	PreSerialize	(inherited from ConfigurationElement)
	Reset	(inherited from ServiceModelExtensionElement)
	ResetModified	(inherited from ConfigurationElement)
	SerializeElement	(inherited from ServiceModelExtensionElement)
	SerializeToXmlElement	(inherited from ConfigurationElement)
	SetPropertyValue	(inherited from ConfigurationElement)
	SetReadOnly	(inherited from ConfigurationElement)
	Unmerge	(inherited from ConfigurationElement)

Explicit Interface Implementations

Name	Description
System.ServiceModel.Description.IServiceBehavior.AddBindingParameters	
System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior	
System.ServiceModel.Description.IServiceBehavior.Validate	

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.CopyFrom Method

Copies the transaction behavior from the specified extension element to the current instance.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides Sub CopyFrom ( _  
    from As ServiceModelExtensionElement _  
)
```

C#

```
public override void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

C++

```
public:  
virtual void CopyFrom (  
    ServiceModelExtensionElement^ from  
) override
```

J#

```
public void CopyFrom (  
    ServiceModelExtensionElement from  
)
```

JScript

```
public override function CopyFrom (  
    from : ServiceModelExtensionElement  
)
```

Parameters

from

the transaction behavior to copy.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.AddBindingParameters Method

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

endpoints

bindingParameters

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.ApplyDispatchBehavior Method

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration
```

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.System.ServiceModel.Description.IServiceBehavior.Validate Method

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

C#

C++

J#

JScript

Parameters

serviceDescription

serviceHostBase

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior Properties

Public Properties (see also [Protected Properties](#))

	Name	Description
	BehaviorType	Overridden. Gets the type of behavior for the inbound transaction.
	ConfigurationElementName	(inherited from ServiceModelExtensionElement)
	ElementInformation	(inherited from ConfigurationElement)
	LockAllAttributesExcept	(inherited from ConfigurationElement)
	LockAllElementsExcept	(inherited from ConfigurationElement)
	LockAttributes	(inherited from ConfigurationElement)
	LockElements	(inherited from ConfigurationElement)
	LockItem	(inherited from ConfigurationElement)
	TransactionIsolationLevel	Gets or sets the transaction isolation level.
	TransactionTimeout	Gets or sets the transaction timeout level.

Protected Properties

	Name	Description
	ElementProperty	(inherited from ConfigurationElement)
	EvaluationContext	(inherited from ConfigurationElement)
	Item	Overloaded. (inherited from ConfigurationElement)
	Properties	(inherited from ConfigurationElement)

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.BehaviorType Property

Gets the type of behavior for the inbound transaction.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration  
  
Public Overrides ReadOnly Property BehaviorType As Type
```

C#

```
public override Type BehaviorType { get; }
```

C++

```
public:  
virtual property Type^ BehaviorType {  
    Type^ get () override;  
}
```

J#

```
/** @property */  
public Type get_BehaviorType ()
```

JScript

```
public override function get BehaviorType () : Type
```

Property Value

Returns a [Type](#) containing the behavior type.

Thread SafetyAny public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small

Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.TransactionIsolationLevel Property

Gets or sets the transaction isolation level.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

'Declaration

```
<ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue:=IsolationLevel.ReadCommitted)> _
Public Property TransactionIsolationLevel As IsolationLevel
```

C#

```
[ConfigurationPropertyAttribute("transactionIsolationLevel", DefaultValue=IsolationLevel.ReadCommitted)]
public IsolationLevel TransactionIsolationLevel { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionIsolationLevel", DefaultValue=IsolationLevel::ReadCommitted)]
public:
property IsolationLevel TransactionIsolationLevel {
    IsolationLevel get ();
    void set (IsolationLevel value);
}
```

J#

```
/** @property */
public IsolationLevel get_TransactionIsolationLevel ()

/** @property */
public void set_TransactionIsolationLevel (IsolationLevel value)
```

JScript

```
public function get TransactionIsolationLevel () : IsolationLevel

public function set TransactionIsolationLevel (value : IsolationLevel)
```

Property Value

Returns a [IsolationLevel](#) containing the isolation level.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions);

The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)

SqlAdapterInboundTransactionBehavior.TransactionTimeout Property

Gets or sets the transaction timeout level.

Namespace: Microsoft.Adapters.Sql

Assembly: Microsoft.Adapters.Sql (in microsoft.adapters.sql.dll)

Syntax

VB

```
'Declaration

<ConfigurationPropertyAttribute("transactionTimeout")> _
Public Property TransactionTimeout As TimeSpan
```

C#

```
[ConfigurationPropertyAttribute("transactionTimeout")]
public TimeSpan TransactionTimeout { get; set; }
```

C++

```
[ConfigurationPropertyAttribute(L"transactionTimeout")]
public:
property TimeSpan TransactionTimeout {
    TimeSpan get ();
    void set (TimeSpan value);
}
```

J#

```
/** @property */
public TimeSpan get_TransactionTimeout ()

/** @property */
public void set_TransactionTimeout (TimeSpan value)
```

JScript

```
public function get TransactionTimeout () : TimeSpan

public function set TransactionTimeout (value : TimeSpan)
```

Property Value

Returns a [TimeSpan](#) containing the transaction timeout level.

Thread Safety Any public static (**Shared** in Visual Basic) members of this type are thread safe. Any instance members are not guaranteed to be thread safe.

Platforms

Development Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the

Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

Target Platforms

The 32-bit and 64-bit editions of Windows Server 2003 R2 (except the Web Editions); The 32-bit and 64-bit editions of Windows Server 2003 with SP2 (except the Web Edition); The 32-bit and 64-bit editions of Windows Server 2008 (except the Web Editions); The 32-bit and 64-bit editions of Windows Vista Ultimate, Enterprise, and Business editions; The 32-bit and 64-bit editions of Windows XP Professional with SP3; Windows Small Business Server 2003 with SP1 or SP2; Windows Small Business Server 2008

See Also

Reference

[SqlAdapterInboundTransactionBehavior Class](#)

[SqlAdapterInboundTransactionBehavior Members](#)

[Microsoft.Adapters.Sql Namespace](#)