





## Reading Sample

*This sample chapter details configuring SAP Plant Connectivity to connect to supervisory control and data acquisition, data historian, and legacy plant automation systems as well as configuring the data server to connect external databases. The chapter also introduces message services, walking through the different settings, creating processing rules and cleanup rules, reprocessing failed messages, and handling successfully processed messages. It also explains how message services are used for SAP S/4HANA and MES integration with SAP MII.*

-  **“Managing Data from External Systems”**
-  **Contents**
-  **Index**
-  **The Authors**

Saha, Jash, Mukherjee, Malakar, Bhattacharjee

### Implementing SAP MII

705 pages | 01/2023 | \$99.95 | ISBN 978-1-4932-2342-8

 [www.sap-press.com/5610](http://www.sap-press.com/5610)

## Chapter 3

# Managing Data from External Systems

*This chapter explains the integration aspects of SAP MII, including how to connect to external manufacturing plant floor systems using SAP Plant Connectivity and data servers, and how to synchronize the plant floor with other parts of the enterprise using message services.*

In the previous chapter, we discussed the installation and configuration activities of SAP MII, along with server administration and required user management in SAP MII. You also learned about content management and transport using SAP NetWeaver Development Infrastructure (NWDI). In this chapter, we'll discuss required configuration in SAP MII to connect to different external data sources, such as SQL databases, LIMS, MES, SCADA, historians, and other systems to query and update data for manufacturing.

This chapter details the configurations in SAP Plant Connectivity to connect to SCADA/historians and legacy plant automation systems and the data server configurations to connect external databases. The chapter also introduces message services, walking through the different settings, creating processing rules and cleanup rules, reprocessing failed messages, and handling no rule and successfully processed messages. It explains how they're used for SAP S/4HANA and MES integration with SAP MII.

### 3.1 Using SAP Plant Connectivity

SAP Plant Connectivity is an adaptor component which is used to provide connectors for various types of shop floor automation systems, such as plant data historians, supervisory control and data acquisition (SCADA), distributed control system (DCS), open platform communications (OPC) servers, programmable logic controllers (PLCs), machines, and so on. There are both standard adaptors such as OPC, OLEDB, File, Socket, and Modbus, as well as proprietary adaptors such as IP21, OSIPI, iHistorian, Citect, and more to connect to specific manufacturing automation and operation management systems as well as directly to machines, devices, and control systems. SAP Plant Connectivity is usually used with SAP MII to integrate with shop floor automation systems to bidirectionally exchange data, but it can also be used directly with other systems such as SAP S/4HANA, SAP Business Technology Platform (SAP BTP), SAP Event Stream Processor, and SAP Extended Warehouse Management (SAP EWM) to integrate those with the shop floor

automation systems and machines. Based on operational data generated from machines, SAP Plant Connectivity can trigger a notification for the destination system and can perform particular tasks. SAP Plant Connectivity also provides enhanced notification functionality where the source system is expecting immediate acknowledgement from SAP Plant Connectivity without involving the destination system.

SAP Plant Connectivity provides different types of connectors to connect to different types of systems and devices. It can connect with equipment and control systems (PLCs) via Socket or Modbus connectors. For equipment and devices providing data directly, such as weighing scales, typically a convertor such as an RS232 TCP/IP ethernet convertor may need to be used to convert a serial connection to ethernet by providing streaming data through the TCP/IP socket. For equipment and devices connected with a control system (PLC) Modbus connector, SAP Plant Connectivity may be used to bidirectionally exchange the data through a Socket connector. Most of the connectors available in SAP Plant Connectivity provide query and notification support, by which they can query the data from the source from SAP MII or can get a notification message in SAP Plant Connectivity (which can be sent to any other destination system) when there is a change in value in the source system or a formula/expression is satisfied based on the data in the source system. SAP Plant Connectivity integrates with equipment and devices via source system and agent configurations created in it, based on the connector types used. Ultimately, SAP Plant Connectivity needs a system to which the data can be sent or can be queried from. Now, let's discuss source systems, destination systems, agents, and notifications in detail in the subsequent sections.

#### Example

In this chapter, we're considering a scenario where one piece of shop floor equipment is generating quality-related data in a file format. SAP Plant Connectivity is reading the source folder where the equipment is placing the quality-data-related file. Next, SAP Plant Connectivity is sending a notification to the destination system (e.g., SAP MII) with the file content, and SAP MII is processing those data for further use.

### 3.1.1 Source System

In SAP Plant Connectivity, a source system specifies a system from which data needs to be queried. You can configure multiple source systems in the SAP Plant Connectivity management console. (After successful installation of SAP Plant Connectivity you can open the SAP Plant Connectivity management console which is explained in Chapter 2, Section 2.3.) You can find the SAP Plant Connectivity management console from the Windows start menu by selecting **SAP Plant Connectivity • Management Console**. From there, you'll get the source data. Agents in SAP Plant Connectivity are responsible for

establishing the data flow between the data source and SAP Plant Connectivity. A source system transforms the server data in the data source into a tag-based data structure that can be browsed. It also converts the protocol-specific data types to the .NET data types used in SAP Plant Connectivity. SAP Plant Connectivity provides a list of source systems that can be used to connect with different types of data sources, as shown in Table 3.1.

Source System Type	Data Source Used
Asset Framework source system	OSIsoft PI Asset Framework, an integral part of the OSIsoft PI system.
Citect source system	Used to connect with CitectSCADA system, a control system for the monitoring and control of production plants.
File system source system	Used to monitor a specific local or remote directory of a system for data.
IP21 source system	Used to establish a connection to the Aspen InfoPlus.21 database.
Modbus source system	Used to establish a connection between SAP Plant Connectivity and PLCs or other devices.
MQTT source system	Used to establish a connection to an MQTT broker (MQTT server).
ODBC source system	Used to establish a connection to any database management system using the Open Database Connectivity (ODBC) interface.
OLEDB source system	Used to establish a connection to an OLEDB-based data source and to execute database queries.
OPC A&E source system	Used to establish a connection to OPC alarms and events server.
OPC DA source system	Used to establish a connection to OPC data access server.
OPC HDA source system	Used to establish a connection to OPC historical data access server.
OPC UA source system	Used to establish a connection to OPC unified architecture.
OSIsoft PI source system	Used to establish a connection to OSIsoft PI system.
Proficy Historian source system	Used to establish a connection to GE FANUC Proficy Historian server.

Table 3.1 SAP Plant Connectivity Source Systems

Source System Type	Data Source Used
Socket source system	Used to establish a connection to a TCP/IP-based socket server.
Timer source system	A source system which changes its value periodically at specified times is treated as a timer. You can use timers to trigger actions using SAP Plant Connectivity at configurable, usually periodic, times.

Table 3.1 SAP Plant Connectivity Source Systems (Cont.)

Now you know about the available source system types in SAP Plant Connectivity and usage of those source systems. Next, let's discuss how to create a source system in SAP Plant Connectivity using an available source system type. Let's consider creation of a file system source system. In the SAP Plant Connectivity management console, you need to click the **Add Source System** button in the **Source Systems** panel. The **Add Source System** window will appear, where you need to select **File System Source System** in the **Source System Type** dropdown field. You need to enter a name and description for the source system. Then you can click the **OK** button, and the source system will be available in the **Source Systems** panel. Now click the source system from the panel and perform the required configuration in the **Settings**, **Authentication**, and **Reliable Connection** tabs, as shown in Table 3.2.

Field	Purpose
<b>Folder to Monitor</b>	You need to enter the folder path from where the existing file will be checked. For example, <i>C:\Users\Myfolder\Doc</i> .
<b>Monitoring Frequency</b>	You can specify how often (in milliseconds) the folder is to be checked for files.
<b>Processing Order</b>	You can select available processing options from the dropdown, such as <b>Random (Optimal Speed)</b> , <b>Oldest Files First</b> , or <b>Newest Files First</b> .
<b>File Mask</b>	You can enter the search criteria for the files that are relevant for processing. The default value is <i>"**"</i> .
<b>Pass File Contents to Destination</b>	If you select the checkbox, then the file content will be sent to a destination system; otherwise, only the file name is sent to the destination system.
<b>Notify Only If File Is Unlocked</b>	If the checkbox is checked, then SAP Plant Connectivity sends a notification message if the read-only indicator for a file is not set.

Table 3.2 SAP Plant Connectivity File Source System Configuration

Field	Purpose
<b>Ignore Unavailability of UNC Folders</b>	If you configure a network folder with a Universal Naming Convention (UNC) path and the folder isn't accessible during runtime, then the agent gets an error status. If you check the <b>Ignore Unavailability of UNC Folders</b> checkbox, then SAP Plant Connectivity ignores this error and continues monitoring as soon as the folder is available again.
<b>File Type</b>	You can select the file type from the available dropdown options, such as <b>Text</b> and <b>Binary</b> .
<b>Text Encoding</b>	You can select text encoding from available list, then the contents of the text files are read from the source folder and the content is converted into a Unicode character string before sending to the destination systems.
<b>Test File and Sample Text</b>	In this field, you can use a text file to test the chosen text encoding. The file contents are displayed in the <b>Sample Text</b> field.
<b>Action</b>	This specifies how to handle the file after successfully reading it from the source folder. You can select a suitable option from the list, such as <b>No Action</b> , <b>Rename</b> , <b>Move</b> , <b>Copy</b> , or <b>Delete</b> .
<b>Destination Folder</b>	This indicates the destination folder to which the file is to be moved if the action is set to <b>Move</b> or <b>Rename</b> .
<b>Rename Mask</b>	In this field, you can specify the new name of the file.
<b>Authentication Required</b>	You can check this checkbox if authentication is required to access the folder in order to access the source file.
<b>Domain</b>	You can specify the Windows domain here.
<b>User Name</b>	You can specify a username for authentication.
<b>Password</b>	You can specify a password for authentication.
<b>Max. Number of Retries</b>	You can specify a maximum number of retries for reliable connection.
<b>Retry Interval</b>	You can select a retry interval in seconds for reading the file.

Table 3.2 SAP Plant Connectivity File Source System Configuration (Cont.)

After entering the required values for configuration, as shown in Figure 3.1, you can save the configuration. In this section, you learned about the process to add and configure source systems. In the next section, we'll discuss destination system configuration in SAP Plant Connectivity.

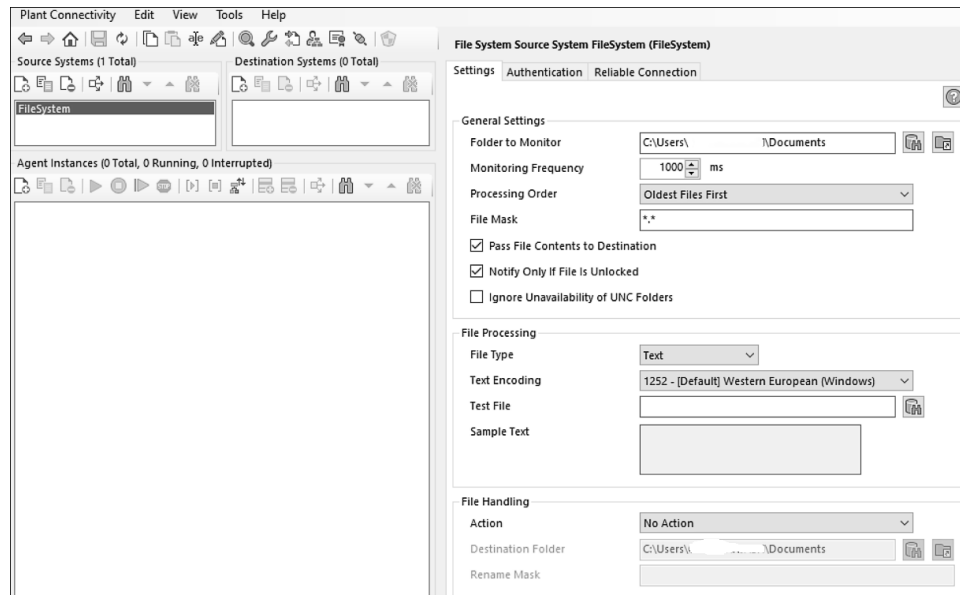


Figure 3.1 Configuring File System Source System

### 3.1.2 Destination System

In SAP Plant Connectivity, the destination system is the system to which the notification is sent based on the condition and notification process. In the **Destination Systems** panel, you can add multiple destination systems and send notifications to appropriate destination systems. SAP Plant Connectivity provides a list of different types of destination systems, as shown in Table 3.3.

Destination System Type	Purpose of Destination System Type
<b>MQTT Destination System</b>	The MQTT destination system is used to send MQTT messages to an MQTT broker. It allows publishing data that has been transferred to SAP Plant Connectivity to an MQTT broker.
<b>Multiple Call Destination System</b>	This is used to call another destination system in a configurable sequence. A multiple call destination (MCD) system also allows other functions that can be combined with each other.
<b>ODBC Destination System</b>	Using an ODBC destination system, you'll be able to connect an ODBC database to SAP Plant Connectivity and you can send notification messages to this database. SAP Plant Connectivity supports connecting to the SAP HANA database and MS SQL Server ODBC database.

Table 3.3 Destination System Type

Destination System Type	Purpose of Destination System Type
<b>OPC UA Destination System</b>	The OPC UA destination system allows you to connect an OPC UA server. You can use OPC UA destination systems for tag-based notifications and method notifications.
<b>Query Destination System</b>	This allows you to retrieve current tag values from the source system of an agent instance or to store tag values to the source system of an agent instance. The agent instance can be the same agent instance from which the query destination system is called, or it can be another agent instance available in SAP Plant Connectivity.
<b>RFC Destination System</b>	This allows you to connect SAP NetWeaver, OPC data access, and SAP EWM.
<b>Simulation Destination System</b>	The simulation destination system is used for testing the notification process very quickly. It's used for testing by sending notification messages to a specific folder.
<b>Universal Web Service Destination System</b>	The universal web service destination system allows you to send notification messages to web services (WSDL), OData services, and RESTful web services. You can also send notifications from SAP Plant Connectivity to SAP MII using a universal web service destination system.

Table 3.3 Destination System Type (Cont.)

To add a destination system in the **Destination Systems** panel, you need to click the **Add Destination System** button in the **Destination Systems** panel. The system will show a new window to add a destination system where you need to select the appropriate type of destination system from the **Destination Systems Type** dropdown (e.g., **Universal Web Service Destination System**). Now you can enter a name and description for the destination system in **Name** and **Description** fields. Finally, you can click the **OK** button, as shown in Figure 3.2.

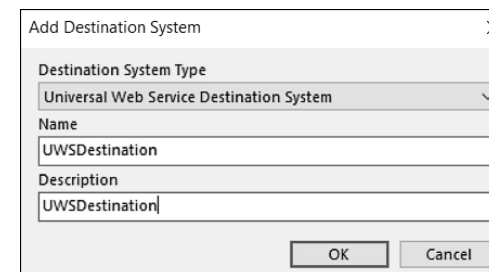


Figure 3.2 Adding SAP Plant Connectivity Destination System

After adding the destination system, you can perform the required configuration for the destination system. We'll use the example of configuring a universal web service destination system to send the file system source system file data to the target SAP MII system. In the **Web Service Settings** tab, choose the description type from the available options, such as **No Service Description** (used for RESTful Web services), **WSDL**, and **OData**. To configure the RESTful web services setting for the SAP MII system, select **No Service Description** value for the **Description Type** field and provide the "https://<server name>:<port number>/XMII/Runner" URI in the **Endpoint URI** field, as shown in Figure 3.3.

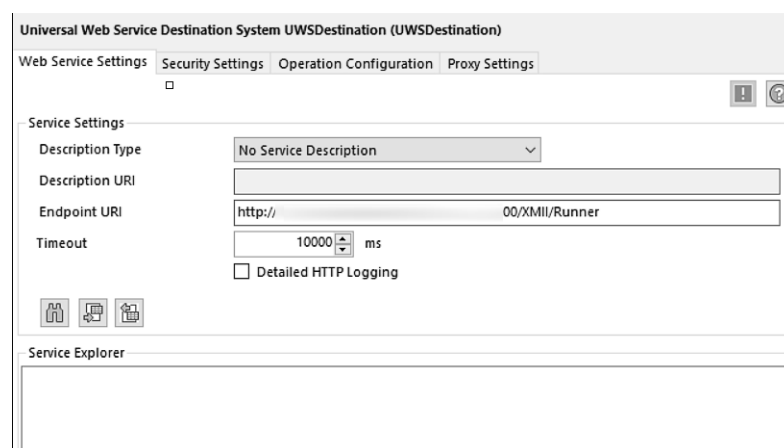


Figure 3.3 Configuring Endpoint URI for SAP MII

On the **Security Settings** tab, you can select the appropriate authentication method and perform the required parameter configurations. On the **Operation Configuration** tab, you can select **POST** as the method. Then move to the **Templates · Request** tab, where you can enter the template, as shown in Figure 3.4. On the **Request Message Configuration** tab, you can find the input variables that appear from values in the curly brackets of the template request. The system interprets the values inside the curly brackets as input variables, which you can find in the **Request Message Configuration** tab. You can assign a **True** or **False** value to the **inIsAsync** variable, which indicates whether the SAP MII transaction can be processed asynchronously. You can set a value of **NEVER**, **ALWAYS**, or **ONERROR** for the **inPersistence** variable, which is used to specify the persistence control of the SAP MII transaction.

In the **inTransaction** variable, you can specify the name of SAP MII transaction—for example, Default/Chandan/Utility/BLS\_PCoIntegration. You can specify the first SAP MII transaction parameter name in the **inParameter1** variable and you can also specify the value of the first parameter in the **inValue1** variable.

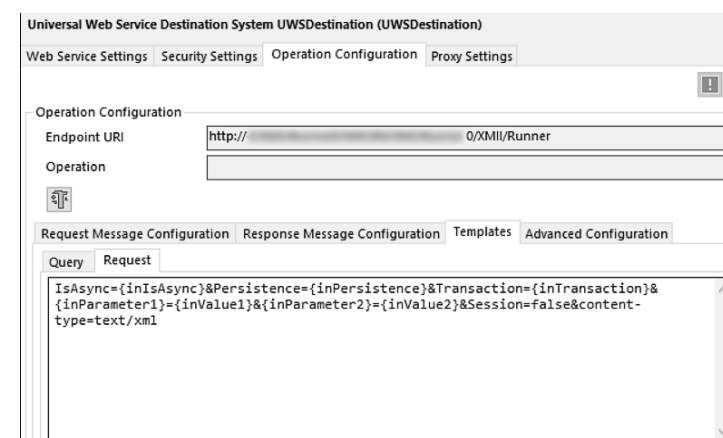


Figure 3.4 Operation Configuration for Universal Web Service Destination System

### 3.1.3 Agent Instance

In SAP Plant Connectivity, the agent instance plays a key component role that's used to establish the connection between the source system and SAP Plant Connectivity. It also helps to enable processing and forwarding messages to the destination system. An agent instance is created based on one source system; it's also possible to create an agent instance without a source system, in which case the agent instance can be used as an OPC UA (unified architecture) server or as a web server. Each agent instance is executed as a standalone windows service, and you can run multiple agent instances at the same time, each of which is executed independently. Now let's see how to create an agent instance in SAP Plant Connectivity.

To create an agent instance, first click the **Add Agent Instance** button under the **Agent Instances** panel. The system will display a popup window where you need to select the source system from available dropdown list. Enter the instance name and instance description, then click the **OK** button. The agent instance will be available under the **Agent Instances** panel. Next you need to perform the required configurations for the created agent instance, as we'll discuss in the following sections.

#### Host

Before starting the agent instance, you need to establish the settings for the host in the **Host** tab of the agent instance. You won't be able to change the configuration in the **Host** tab if the agent instance is in a running state. In the **Host** tab, select a **Log Level** from the available values, such as **Off**, **Critical**, **Error**, **Warning**, **Information**, and **Verbose**. Selecting the **Run Agent Instance as an Executable** checkbox allows the agent instance to run as a normal windows process with the rights of the logged-in user account. It runs as the user logged on in Windows and it stops if the user logs off. If the

**Run Agent Instance as an Executable** checkbox isn't checked, then the agent instance runs as a Windows service.

If the agent instance needs to be run as a Windows service, then you need to specify a service username, password, and start mode in the **Service User Name**, **Service User Password**, and **Service Start Mode** fields. The **Maintained Dependent Services** button is used to maintain any dependent service that can be started before the agent instance is started. The **Startup Timeout** field allows you to set the maximum time in minutes taken to start the agent instance. If the agent instance isn't started by that time, then the start process is terminated. In the **Starting Group** field, you can assign an agent instance to an existing starting group or you can create a starting group using the **Create New Starting Group** button and assign it to an agent instance, as shown in Figure 3.5. Using a starting group, you can start and stop multiple agent instances together that belong to the same starting group. If you want to stop an agent instance individually, then you can select the **Agent Instance Must Be Stopped Individually** checkbox.

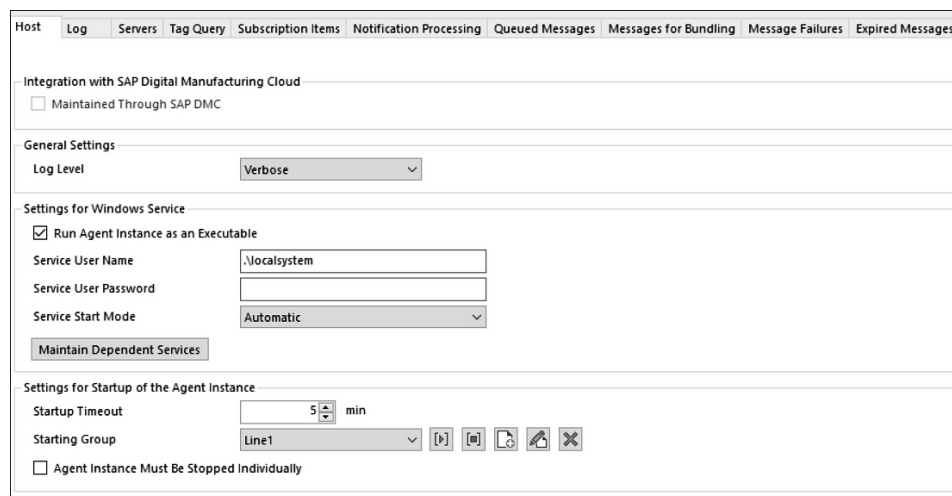


Figure 3.5 Configuring SAP Plant Connectivity Agent Instance Host

### Log

In the **Log** tab, you can search the log entries for the agent instance. SAP Plant Connectivity writes separate logs for individual agent instances, and log data are saved in the Windows event log. You can configure the required fields in the **Log** tab to restrict the search for log data, as shown in Table 3.4.

Field Name	Purpose
Start Date	Start date and time for searching log entries.
End Date	End date and time for searching log entries.

Table 3.4 Agent Instance Log Configuration

Field Name	Purpose
Filter Level	Refers to the log type that will be displayed in a search. Available values for filter level are <b>Critical</b> , <b>Error</b> , <b>Warning</b> , <b>Information</b> , and <b>Verbose</b> .
Number of Entries	Lets you set the number of log entries to display. Available values are <b>1000</b> , <b>2000</b> , <b>5000</b> , and <b>All Entries</b> .

Table 3.4 Agent Instance Log Configuration (Cont.)

After selecting the date and filter conditions, click the **Refresh** button. The log entries are displayed in a table, in descending order based on the time of the logged events. You can delete all log entries by clicking the **Delete All Log Entries** button. Using the **Export to CSV File** button, you can export searched log entries in CSV file into a folder as per your selection. By clicking the **Manage Windows Event Log** button, you can start the Windows Event Viewer for the Windows log assigned to the agent instance.

At the bottom of the log entries table, you can use the **Search For** field to select displayed log entries that match specific search criteria. The available values in the **Search For** field are **Messages**, **Event ID**, **Event Type**, and **Source**. You can select one of the values from the field and click the **Execute Search** button to select appropriate log entries based on the search criteria. **First Selected Log Entry**, **Previous Selected Log Entry**, **Next Selected Log Entry**, **Last Selected Log Entry**, **Select Log Entries with Same Thread IDs**, and **Copy Selected Log Entries to Clipboard** buttons are also available to perform manual functions on selected individual or multiple log entries.

### Servers

In the **Servers** tab of agent instance, you can configure various server types, which are used for different query interfaces. The available server types you can select for agent instances are listed in Table 3.5.

Server Type	Required Settings
SAP MII Query Server	Used where you're using SAP MII 12.2 or higher and you want to connect with SAP Plant Connectivity for processing the tag query or PCo query. To perform the server configuration for an agent instance, you need to check the <b>SAP MII Query Server</b> checkbox in the <b>Available Server Type</b> list in the <b>Servers</b> tab. You can specify the port for connection in SAP MII in the <b>Port</b> field. In the <b>Security</b> field, you can choose an available value, such as <b>Unsecured Connection</b> or <b>Connection with TLS</b> . In the <b>Server Certificate</b> field, you can specify the certificate used by SAP Plant Connectivity to establish a secure connection with SAP MII.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List

Server Type	Required Settings
<b>SAP MII Query Server (Before 12.2)</b>	Used where you're using an SAP MII system version lower than 12.2 and you want to connect with SAP Plant Connectivity for processing a tag query or PCo query. To perform the server configuration for an agent instance, you need to check the <b>SAP MII Query Server</b> checkbox in the <b>Available Server Type</b> list in the <b>Servers</b> tab. You can specify the port for connection in SAP MII in <b>Port</b> field (the same port has to be specified for the PCoConnector data server configured in SAP MII). One port is to be used for one agent instance only. In the <b>Include Quality</b> field, you can select one value from the available options, such as <b>No Quality Information</b> , <b>Quality Information as String</b> , <b>Numeric Quality Information</b> , and <b>Quality Information as String and Numeric</b> . The value in the <b>Include Quality</b> field indicates how quality information is to be provided for the measurement values that are to be transmitted. There are two checkboxes for UDS compatibility: <b>Calculate Minimum and Maximum for Intervals without Value Changes</b> and <b>Assign Statistical Values to End of Interval</b> . The latter indicates that the calculated statistical values are always assigned to the end of the evaluation interval.
<b>SAP NW RFC Server</b>	If you need to connect SAP Plant Connectivity with SAP S/4HANA for query processing, then you can set up an SAP NetWeaver RFC server in the <b>Servers</b> tab of the selected agent instance. To establish the connection, you need to fill in the following fields: <b>Registration Using Message Server</b> , <b>SNC</b> , or <b>Tracing and Repository</b> . You need to provide values to establish the connection, in the fields <b>Program ID</b> , <b>SAP Gateway Host</b> , <b>SAP Gateway Service</b> , <b>SAP Router</b> , and so on. You can test the connection by clicking the <b>Test Registration</b> button.
<b>SAP ODA RFC Server</b>	If you need to achieve Open Platform Communications (OPC) Data Access functionality in an SAP S/4HANA system then you can select <b>OPC Data Access RFC Server</b> in the <b>Servers</b> tab of the agent instance. The setting is only possible in connection with an OPC Data Access agent. Using the <b>Test Registration</b> button, you can test the connection. You need to enter the values for <b>Program ID</b> , <b>SAP Gateway Host</b> , and <b>SAP Gateway Service</b> ; when using load balancing, you need to enter values for <b>SAP Message Server Host</b> , <b>SAP System ID</b> , and <b>SAP Logon Group</b> .
<b>SAP EWM RFC Server</b>	Required when you want to connect an SAP EWM system with SAP Plant Connectivity to send telegrams. You need to provide required information to establish connection with the SAP EWM system. Using the <b>Test Registration</b> button, you can test the registration; you need to configure required fields like <b>Program ID</b> , <b>SAP Gateway Host</b> , <b>SAP Gateway Service</b> , or <b>SAP Message Server Host</b> , <b>SAP Logon Group</b> , and <b>SAP System ID</b> if you're using load balancing.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List (Cont.)

Server Type	Required Settings
<b>WebSocket Server</b>	Used when you want SAP Plant Connectivity to act as a WebSocket server for web applications. The notification process and query process are supported by the WebSocket server. In the notification process, the events provided by the source system can be displayed in the UI. In the query process, the client can send messages to the SAP Plant Connectivity WebSocket server. You need to enter required details for configuring WebSocket settings, such as <b>Protocol</b> , <b>Port</b> , <b>Server Certificate</b> , <b>Message Format</b> , and so on. You can also maintain the destination system by clicking the <b>Created Destination System</b> button. Using the <b>Delete Destination System</b> button, you can delete a destination also. The status of the created destination system will be displayed in the <b>Status</b> field.
<b>OPC UA Server</b>	Used if you want to run an agent instance as an OPC UA server. In the <b>OPC UA Server Settings</b> tab, you can define the server endpoint. In the <b>Security Configuration</b> tab, you can configure secure connections and user authentication for OPC UA servers. In the <b>Server Method Definitions</b> tab, you can add new method definitions to the server.
<b>PCo Web Server</b>	If you want to run an SAP Plant Connectivity agent instance as a web server, then you can use this checkbox to configure the web server. In the <b>PCo Web Server Settings</b> tab, you can add an end server endpoint, and you can define a web server method in the <b>Web Server Method Definitions</b> tab.

Table 3.5 SAP Plant Connectivity Agent Instance Server Type List (Cont.)

### Tag Query

In the query process, required tags can be specified in the **Tag Query** tab. You can establish the cache settings in the **Tag Query** tab with the **Cache Mode**, **Mask**, and **Alias** fields. The **Cache Mode** field provides four options you can choose from per your requirements:

- **Access to Data Source Only**  
Allows you to retrieve directly from the data source for queries.
- **Access to Cache, to Data Source as Required**  
Allows you to check the cache first to see whether there are tags and metadata. If not, then the tags are retrieved from the data source and added to the cache.
- **Access Using Aliases Only**  
Used if you previously created an alias for the source system.
- **Access to Cache Only**  
Used to query all tags from the data source that are stored in the cache.



### Subscription Items

In the **Subscription Items** tab, you can subscribe to tags from the source system; these subscriptions can be used for notification processes by the agent instance. You can also change subscriptions or delete subscription items from the list. You can browse multiple tags from the source system by clicking the **Browse for Tags** button in the **Subscription Items** tab. You can also add individual tags by clicking the **Add New Subscription Item Manually** button. The subscription items will be displayed in a table inside the **Subscription Items** tab, and the table contains fields like **Name**, **Source**, **Deadband**, **Only Changes**, and **Data Type**, as described in Table 3.6.

Field Name	Purpose
<b>Name</b>	This is the name of the tag.
<b>Source</b>	This is the tag name in the namespace of source system.
<b>Deadband</b>	In the SAP Plant Connectivity notification process, deadband refers the range of allowed variance in values and is used to filter out disallowed measurement values. For example, let's say you have specified a deadband value of "2" in the <b>Subscription Items</b> tab. The last tag value was 52 and next tag value is 54.5. The difference between these two values is 2.5 (53.5 – 52); this is higher than the specified deadband value 2. As such, SAP Plant Connectivity will send a notification to the destination system.  If the difference between the current tag value and the next tag value had been less than specified deadband value then SAP Plant Connectivity would not have sent a notification to destination system.
<b>Only Changes</b>	For a tag, if you select the <b>Only Changes</b> checkbox then SAP Plant Connectivity sends a notification to the destination system if the tag value is changed compared to last value of the same tag; otherwise, no notification will be sent.
<b>Data Type</b>	Refers to the data type of the tag. The data type of the tag is determined automatically during browsing for tags. If the data type isn't supported by SAP Plant Connectivity, then the tag can't be selected during browsing.

Table 3.6 SAP Plant Connectivity Agent Instance Subscription Items

You can also change the attribute of a subscribed tag by selecting the tag's row from the subscription item list and clicking the **Edit Subscription Item** button.

### Notification Processing

In the **Notification Processing** tab, you can perform two types of activities for notification messages: **Notification Message Queue and Dispatch Settings** and **Enhanced Notification Processing**. In the notification process, before sending messages to the destination system, the messages are placed in a queue, which helps avoid blocking the

processing of further notifications from source system. It also helps to start a delivery retry if delivery fails. Individual message queues are created automatically for each agent instance, and queues are deleted when the agent instance is deleted. Different types of queues are created for each agent instance, such as the main queue, queue for message bundling, failed message queue, and expired message queue. In the **Notification Message Queue and Dispatch Settings** section, you can establish the required settings, as explained in Table 3.7.

Field Name	Purpose
<b>Storage Method</b>	Refers to how notification messages can be stored in the queue. The available options are <b>In Memory Only</b> , <b>File System</b> , and <b>Microsoft Message Queuing (MSMQ)</b> .
<b>Keep Messages in Memory</b>	This checkbox selection option is available for the file system storage method only. It's autoselected for the other two storage methods. If this checkbox is selected, then all messages are stored in SAP Plant Connectivity memory.
<b>Resend Failed Messages Automatically</b>	If the checkbox is selected, then the system tries to resend the failed messages automatically. If the checkbox isn't selected, then you must perform manual resending of messages from the <b>Message Failure Tab</b> .
<b>Keep Expired Messages</b>	Selection of this checkbox specifies that expired notification messages are moved to the expired messages queue. You can find the expired messages on the <b>Expired Messages</b> tab. If the messages aren't deleted from the expired messages queue, then that queue grows continuously. If the checkbox isn't selected, then notification messages are deleted automatically from all queues after expiry.
<b>Process Notification Messages Exactly Once in Order</b>	Indicates all notification messages are delivered individually and exactly once in the order in which they were placed in the queue.
<b>Keep Copies of Queued Notification Messages in Journal Queue</b>	Selection of this checkbox specifies that a copy of a notification message is stored in the <b>Journal Messages</b> folder of MSMQ. The system will allow you to select the checkbox if you selected the MSMQ storage message. Keep in mind that this option should be used for testing purposes only; otherwise, it can affect system performance.
<b>Make Queued Notification Messages Recoverable</b>	The system allows you to select this checkbox if you selected the MSMQ storage method. If checked, notification messages are stored on the system hard disk and are available after a Windows restart. Notification messages are saved in the working memory if the checkbox isn't selected.

Table 3.7 SAP Plant Connectivity Agent Instance Notification Processing

Field Name	Purpose
Max. Queued Messages	This field allows you to set the value for the maximum number of messages to be in the message queue; this includes the messages in the <b>Queued Messages</b> tab and in the <b>Message Failures</b> tab.
Max. Dispatch Threads	You can enter the value for the maximum number of threads per CPU core for parallel sending of notifications. The minimum and maximum range is 1 to 250 per CPU core. The value you enter will be multiplied by 8 (because your SAP Plant Connectivity system has eight CPU cores). You need to keep in mind that this setting isn't applicable for the <b>Process Notification Messages Exactly Once in Order</b> option. If the <b>Process Notification Messages Exactly Once in Order</b> checkbox is checked, then the <b>Max. Dispatch Threads</b> field will be disabled.

Table 3.7 SAP Plant Connectivity Agent Instance Notification Processing (Cont.)

In the **Enhanced Notification Processing** section, you can configure the enhancement of the notification process. Based on the selection of available radio buttons, different options will be available for enhancement as described in Table 3.8.

Field Name	Purpose
No Enhanced Notification Processing	You can select the radio button if there is no requirement for enhanced notification.
Destination System Calls with Response Processing	This is the SAP-standard enhanced notification process available with an SAP Plant Connectivity installation. If you select the radio button, then the SAP Plant Connectivity standard enhanced notification is tagged automatically. After selecting the radio button, you need to create the destination system by clicking the <b>Create Destination System</b> button in the <b>Maintain Destination System for Notification Enhancement</b> section.
Automatic Configuration Backup	You can select this option if you want to create a data backup of the configuration. To setup automatic data backup, select the menu path <b>Plant Connectivity • Set Up Automatic Backup</b> . The system will show a popup window where it creates a source system and an agent instance with the name AutoBackupConfig. Then you can set the execution plan for the timer in the <code>_AutoBackupConfig</code> source system and start the agent instance.  In the <b>Notification Processing</b> tab of any agent instance, select the <b>Automatic Configuration Backup</b> radio button. Then, you can create the destination system for this notification processing by clicking the <b>Create Destination System</b> button.

Table 3.8 SAP Plant Connectivity Agent Instance Enhanced Notification Processing

Field Name	Purpose
Automatic Configuration Backup (Cont.)	Next, you need to create a static notification for the agent instance, and in <b>Output</b> tab of the notification, generate the output expression by clicking the <b>Generate Expressions</b> button. In the <b>Destinations</b> tab, you can assign the <b>/Enhanced Notification Processing</b> destination system to the notification.
Customer-Owned Enhancement	Using this option, you can create your own enhancement where you need to create your own dynamic link library (DLL) and enter it in the <b>Dynamic Link Library</b> field by clicking the <b>Browse</b> button. If you have only one class file that implements the enhanced notification processing interface, then that class appears automatically in the <b>Class</b> field by selection of the DLL file.

Table 3.8 SAP Plant Connectivity Agent Instance Enhanced Notification Processing (Cont.)

#### Note

Enhanced notification processing allows you to control and document the data flow flexibly in production in connection with one or more destination systems one after the other in any order—for example, a web service, a RESTful web service, an OData destination, or an ODBC destination. SAP also delivers the standard enhancement destination system calls with response processing for enhanced notification processing. SAP Plant Connectivity allows you to execute one or multiple destination system calls and you can write back the results of the calls to the data source of the agent instance or another agent instance.

#### Queued Messages

In the **Queued Messages** tab, you can find all notification messages currently queued. You need to click the **Refresh** button to get the queued message list in this tab.

#### Messages for Bundling

In **Messages for Bundling** tab, you'll be able to find the messages that are to be bundled by clicking the **Refresh** button. To get the messages in the **Messages for Bundling** tab, you need to configure message bundling on the **Message Delivery** tab of the notifications in the agent instance. In the message bundling concept, you can specify that instead of sending individual messages to a destination system, you'll either send a fixed number of messages in a bundle, or set a maximum time in seconds to wait to accumulate the messages that will be bundled into a single message to be sent to the destination system.

**Example**

Consider a scenario where a high-speed shop floor machine is generating data in second and that same data needs to be sent to SAP MII through notifications. You can bundle the messages per your requirements and configure a fixed number of messages or maximum accumulation time, then we can send the bundled messages to the SAP MII system in a single call instead of making multiple calls to the SAP MII destination system. This can increase system performance.

**Message Failures**

In the **Message Failures** tab, you need to click the **Refresh** button to see the failed messages that haven't been sent successfully even after multiple attempts. You can send one or more messages by clicking the **Resend All** button and you can delete messages by clicking the **Delete All Messages** button.

**Expired Messages**

In the **Expired Messages** tab, you can see the messages that couldn't be sent and that are past the predefined lifetime. You need to click the **Refresh** button to see the message list, and you can delete messages by clicking **Delete All Messages**.

**3.1.4 Notification**

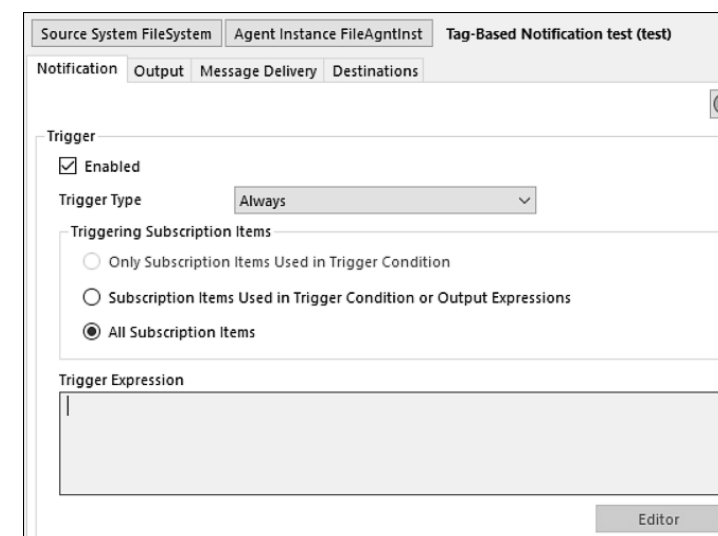
A notification is defined under the agent instance to send message information to the destination system based on the occurrence of a specific event. To add a notification under the agent instance, you need to select the agent instance and click the **Add Notification** button. The system will show the options for notification types such as **Static Notification**, **Notification Template for Remote Subscriptions**, **Versioned Notification**, and **Method Notification**. Static notifications are in the tag-based notification category. Static notifications are treated as normal notifications. The notification template is a copy template for notification messages; you need to create the template for remote subscription. The versioned notification contains one or multiple notifications with different versions, and notification versions can be draft, active, or expiring versions. You can select a method notification option when you want to activate a method of the OPC UA server or SAP Plant Connectivity web server, and you can define which action you want to run if the method is called by a client.

**Example**

Consider the scenario in which an agent instance is created based on a file system source system in SAP Plant Connectivity. Shop floor equipment is generating the quality-related data for a specific product and placing those data in a file. Now the created

notification under same agent instance will send the file content to the destination system (e.g., SAP MII) based on the trigger type and condition which you can find in later in this section. You can send the notification in XML format or you can send the values for specific fields that will be processed in SAP MII.

Once the notification is added under the agent instance, you need to configure the notification, which will send an XML message, defined in the output configuration, to the destination system (e.g., SAP MII). You need to create a notification for the agent instance as shown in Figure 3.6. In the **Notification** tab, you can specify the *trigger expression* for the notification—that is, the condition based on which the notification message will be generated. You can specify **Always** in the **Trigger Type** dropdown, which will generate the notification always whenever there is a change of the value in the source system. In that case, you don't need to specify a trigger expression. You can also select **OnTrue**, **OnFalse**, **WhileTrue**, or **WhileFalse** under **Trigger Type**. **WhileTrue** or **WhileFalse** will send a notification message every time the value changes and the expression evaluates to true or false, respectively, but for **OnTrue** and **OnFalse**, it will send the notification message once when the expression to be evaluated evaluates to true or false, respectively, and won't send it again unless the expression evaluation result changes.



**Figure 3.6** Configuring SAP Plant Connectivity Agent Instance Notification

In the **Output** tab, you can define the information that is to be passed to a destination system. The output values are added based on subscription items, as shown in Figure 3.7. You can add any number of elements in the output message to specify the actual subscription item values from the source system, a timestamp, or any other fixed values. The subscription items can be added in the output expression within single

quotes. Any fixed value can be added within double quotes. Figure 3.7 shows the output configuration for sending file content to the SAP MII system.

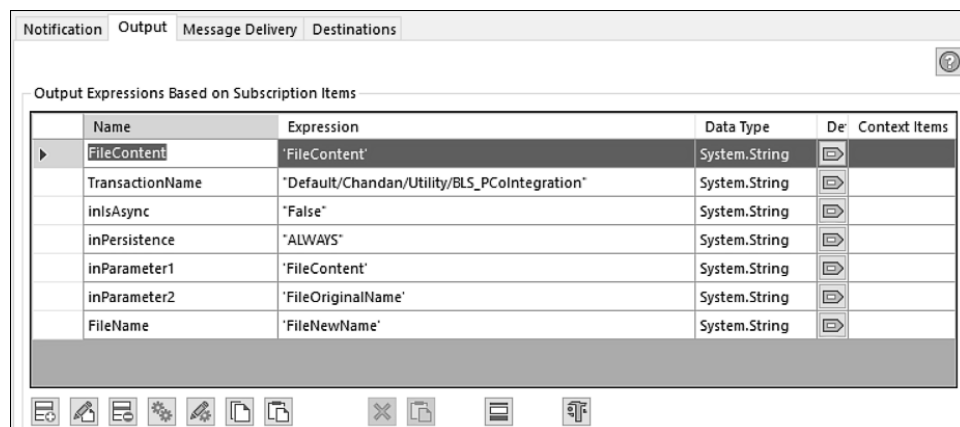


Figure 3.7 Configuring Output for SAP Plant Connectivity Agent Instance Notification

In the **Message Delivery** tab, you can configure how to deliver messages to the destination system. In the **Reliable Connection** section, you can configure values for **Max. Number of Retries** and **Retry Interval (Seconds)**. In the **Failed Message Persistence** section, you can select values for how failed notification messages are to be stored, and you can enter the value in minute for the **Delete Messages After** field. All failed messages older than the value in **Delete Messages After** are to be deleted. In the **Lifetime** section, you can specify how long the messages are to be kept in the queue. For many of these scenarios, the frequency of the data coming from the machine or control system may be very high (e.g., a few hundred messages in a minute), so sending each of them individually to a destination system such as SAP MII may be a huge source of performance overhead. You can configure to bundle the notification messages in the **Message Delivery** configuration tab of the notification, as shown in Figure 3.8. You can specify either a fixed number of messages to bundle or a maximum time in seconds to wait to accumulate the messages, which will be bundled into a single message to be sent to the destination system. Note that message bundling is not supported for enhanced notification processing or query destination.

Finally, you need to perform configuration for the **Destinations** tab to specify the destination where the notification message will be sent. SAP Plant Connectivity can send data to different types of destination systems, as you know from the discussion of the creation of different destination systems in Section 3.1.2. The **Destinations** tab is used to configure a notification and to configure a notification template. To add a destination, you can click the **Add Destination System** icon. The **Add Destination System to Notification** popup will appear, where you can select the destination system that you want to use for notification and enter a **Name** and **Description**. You can add a destination for a Universal Web Service destination system that will be used to send notifications to the

SAP MII system, as shown in Figure 3.9. You can then perform assignment of the destination system input variables to output values.

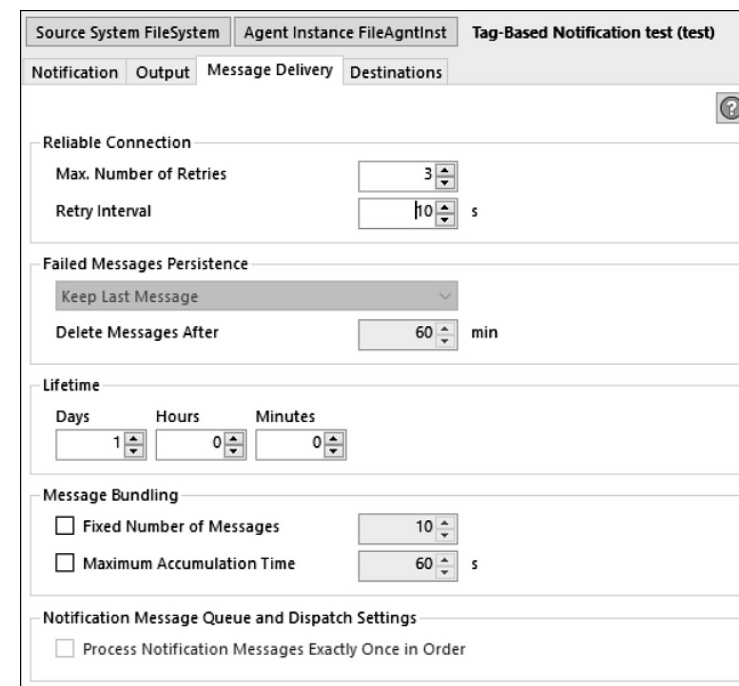


Figure 3.8 Configuring Message Delivery

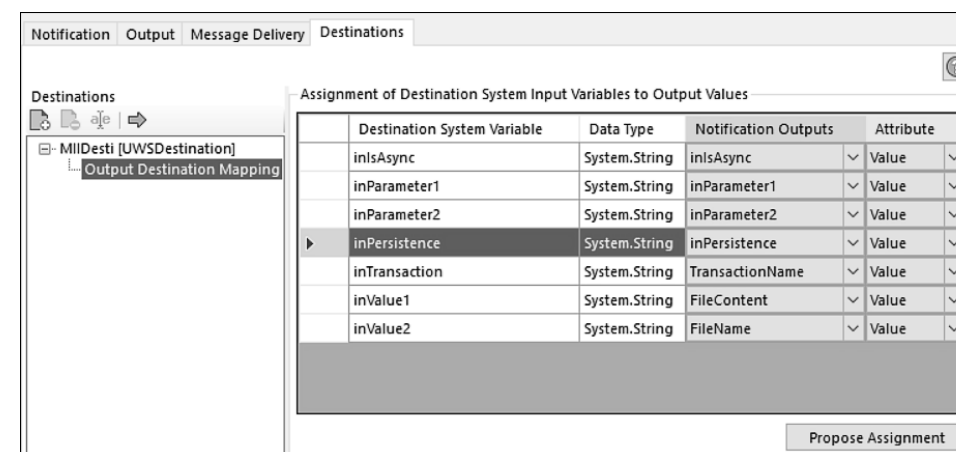


Figure 3.9 Configuring SAP Plant Connectivity Notification Destination

You can also send the notifications to the destination system (e.g., SAP MII) as XML, and SAP MII can process the XML notifications. Hence, you need to define an input parameter (e.g., **InputXML**) of type XML in SAP MII in **Transaction Properties**. Now, the same

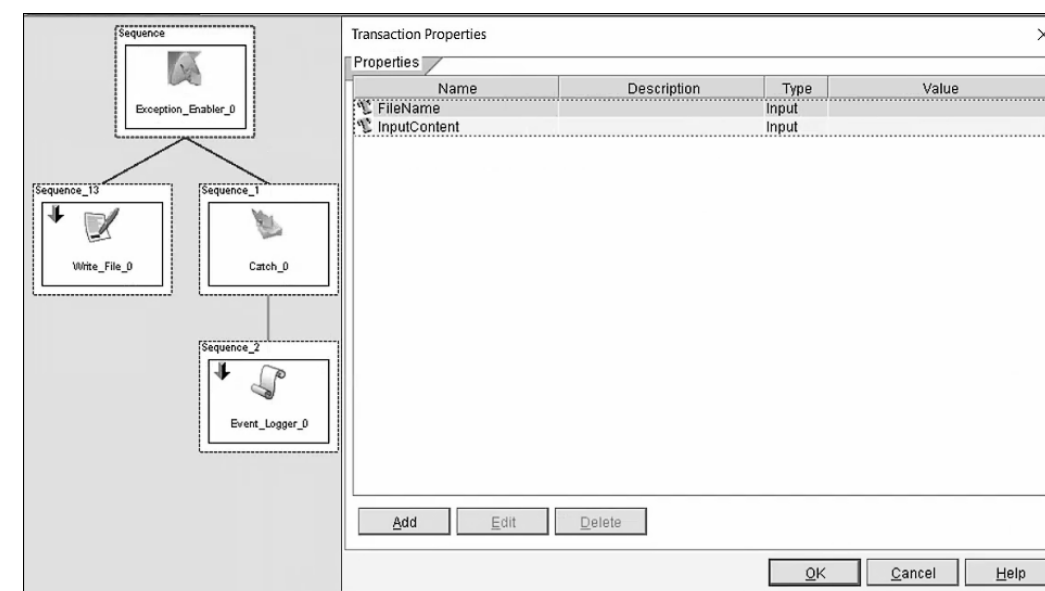
input parameter needs to be mapped to the `inParameter1` variable (e.g., in the **Expression** field you can map `InputXML`) in the **Output** tab of the agent notification. Next, in the **Destination** tab, you can assign `~NotificationOutputInXMLFormat` to the `inValue1` destination system variable. You can find a sample XML notification received in SAP MII from SAP Plant Connectivity in Listing 3.1.

```
<?xml version="1.0" encoding="UTF-8"?>
<NotificationMessage>
  <Header>
    <Name>Test_Notification</Name>
    <Description><![CDATA[Test_Notification]]></Description>
    <Destination>MII_DEST</Destination>
    <CreateDate>2022-10-19T12:46:44</CreateDate>
    <ID>2cd33216-1b4e-45bd-82f6-caf5fe673818</ID>
    <Status>Created</Status>
  </Header>
  <Body>
    <Values>
      <FileContent type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[AHJAAHJS]]></FileContent>
      <TrasactionName type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[Default/Chandan/BLS/BLS_TestPcoFile]]>
</TrasactionName>
      <inIsAsync type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[False]]></inIsAsync>
      <inPersistence type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[ALWAYS]]></inPersistence>
      <inParameter1 type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[InputContent]]></inParameter1>
      <inParameter2 type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[FileName]]></inParameter2>
      <FileName type="xsd:string" quality="None" timeStamp="0001-01-01T00:00:00.000000"><![CDATA[dkjdkjdkjdkj]]></FileName>
    </Values>
  </Body>
  <Faults />
</NotificationMessage>
```

**Listing 3.1** Sample SAP Plant Connectivity XML Notification Received in SAP MII

Once the configuration is complete, you need to start the agent instance from the SAP Plant Connectivity console, which will start the service to listen to the file system source. If you place a file in a monitoring folder, then the system will read the file, and then a notification message will be sent to the destination system for further processing. Starting the agent instance is also required to query the data source from SAP MII.

A sample notification message sent to SAP MII is shown in Figure 3.10. You can develop custom logic using the BLS transactions in SAP MII to process the notification message and initiate further processing to store the data in database, trigger alerts or user actions, or update enterprise systems.



**Figure 3.10** SAP MII Transaction for Notification Message from SAP Plant Connectivity

### 3.1.5 Managing Notifications from SAP MII

In the previous section, you learned about the required configurations in SAP Plant Connectivity for notification. Now you'll learn how notifications are managed in SAP MII. You can directly create and configure notifications in SAP MII and publish the notifications to SAP Plant Connectivity. SAP Plant Connectivity connects to a data source and can handle the notifications published. You can handle notifications for multiple SAP Plant Connectivity installations from a central SAP MII system. Notifications can be paused or modified from SAP MII. SAP MII provides a UI for notification management, which is available in SAP MII by following menu path **Plant Connectivity Integration • Notification Management**. When you click the **Notification Management** link, the system will open a UI with the URL `http://<Host>:<Port>/XMII/PcoManagement.jsp` that contains the **Manage Notifications by SAP Plant Connectivity**, **Manage Notifications by PIC**, and **Browse Notifications** activities, as shown Figure 3.11.

Using notification management functionality in SAP MII, you can create and configure notifications that are processed on the SAP Plant Connectivity side. Based on actions such as create, modify, and publish, the status of the notification is changed in SAP MII and in SAP Plant Connectivity. In SAP MII, when you change an existing notification with the status set to **Current** or **Current-Paused**, then a new version is created. If you

edit a notification with the status set to **Current** from SAP MII, then a new version of the notification with the status set to **Draft** is created in SAP MII and the notification status of **Current** is changed to **Outdated**. In SAP Plant Connectivity, a new version of the notification with the status **Draft** is created and the status of an existing notification is changed to **Expiring**. If you edit a notification with the status **Current-Paused**, then a new version of the notification will be created with status **Draft**. After publishing and activating a notification, the status is changed to **Current-Paused**.

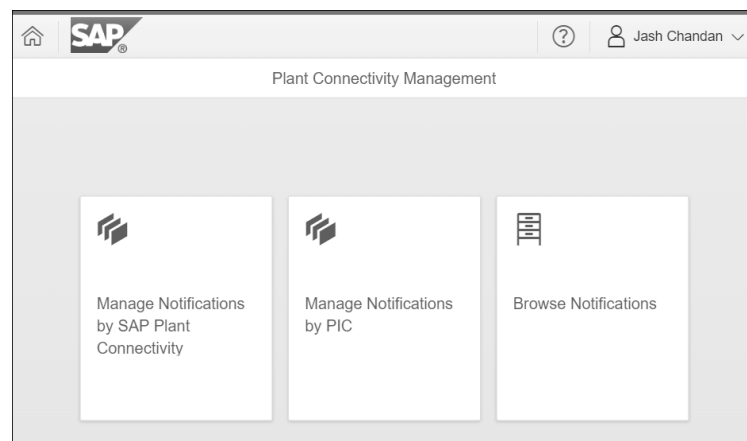


Figure 3.11 SAP MII Notification Management

In SAP MII, you can create a notification and publish it to SAP Plant Connectivity. You can create a notification by browsing tags from SAP Plant Connectivity or the PIC. The initial status of a notification in SAP MII is set to **Draft**, but after publishing the notification to SAP Plant Connectivity the status is set to **Draft** in SAP Plant Connectivity. In the **Browse Notifications** page, you can search by SAP Plant Connectivity data server, notification name, and status. Notification details will be displayed in the **Notifications** table, where you can perform activities like export, import, activate, and pause for selected notifications from the table.

### 3.2 Connecting to External Systems Using Data Servers

Data servers in SAP MII are the most important aspect of manufacturing plant floor connectivity because they provide direct and near-real-time bidirectional access to the manufacturing data present in different systems and applications on the plant floor. Via standard protocols and application interfaces, the data servers provide direct connections to databases or applications in levels 2 and 3 of an enterprise. A considerable list of data connectors is provided in SAP MII. You can configure these as data servers and, using query templates, retrieve or write data to these systems. The list of different connector types available in SAP MII 15.5 is as follows:

- AggregateConnector
- AlarmSuite
- CatalogConnector
- DataSource
- IDBC
- IP21
- IP21OLEDB
- InSQL
- InSQLPCo
- KPIConnector
- MDOConnector
- OLAP
- OLEDB
- OpenConnector
- PCoConnector
- PCoSimulator
- Simulator
- VirtualAlarm
- VirtualCatalog
- VirtualIDBC
- VirtualKPI
- VirtualMDO
- VirtualOLAP
- VirtualPCo
- VirtualTAG
- VirtualXML
- VirtualXacute
- XMLConnector
- XacuteConnector

#### Note

A few notes on some of these connectors:

- Aggregate Connector was introduced to combine multiple query templates in a single data query. But now this can be achieved more easily and flexibly using BLS transactions.
- AlarmSuite Connector was introduced to connect to Wonderware AlarmSuite, which is currently an obsolete application.
- IP21 Connector was introduced to connect to AspenTech InfoPlus.21, which is a real-time information management system for manufacturing processes. It utilizes the Easysoft JDBC/ODBC driver, which you need to deploy to connect to a system Data Source Name (DSN) on the machine where the InfoPlus.21 suite is installed. Now IP21 Connector is the recommended connector to connect to AspenTech InfoPlus.21, which connects via SAP Plant Connectivity.
- XML Connector was introduced to connect to XML data sources using HTTP, but now it's easier to achieve the same end using BLS transactions. Nevertheless, XML Connector can be used to execute any illuminator service in MII or any HTTP service using XML.

Each data server provides connections to specific applications or systems used on the manufacturing plant floor. Any application or system using one of the previously mentioned connectors can be integrated with SAP MII.

The data servers provide different query modes to retrieve different types of data such as current data, historical data, statistical data, and so on. This data varies with the type of data server and is specified when you're configuring the query templates using the data servers. Also, each mode might have one or more query methods, such as AVG, MIN, MAX, and so on, which provide specific data for that specific query mode.

In the following sections, we'll first look at the configuration of a data server in SAP MII, and then discuss many of the different connector types listed earlier.

### 3.2.1 Data Server Configuration

You need to create a separate data server configuration in SAP MII for each plant system you are connecting to. You can create the data server configuration from the **Data Servers** menu under the **Data Services** menu category.

In the **Data Servers** configuration page, shown in Figure 3.12, the existing data servers are shown in a table. The table contains the status, name, and type of each configured data server. You can search for a specific type of data server from the existing data server table by selecting the connector type in the **Connector Type** dropdown field.

To create a new data server configuration, click the **Create** button at the top of table. The new data server configuration screen will appear, where you can enter the server name in **Server Name** field, select a connector type from the **Connector Type** dropdown, and enter a description of a data server in the **Description** field. Next, click the **Finish** button to create your new data server. There are two other buttons available, **Previous** and **Next**, which are enabled based on the type of connector selected. You can copy an existing data server configuration by selecting the existing configuration from the data server table, clicking the **Copy** button, and then entering the name of copied data server. Save the configuration by clicking the **Save** button.

To delete the selected data server configuration, use the **Delete** button from the data server configuration screen. You can edit an existing data server using the **Edit** button and disable a selected data server by clicking the **Disable** button. You can also enable an existing disabled status data server by clicking the **Enable** button. You need to use the **Cancel** button to cancel your changes. You can see the status of all enabled data servers configured in SAP MII from the **Data Services • Connection Status** menu option. The details of a selected connector are displayed at the bottom of the data server table. The details of the connector are displayed in four tabs:

- The **Settings** tab contains the name, connector, connector type, and description information of the data source. It also contains an **Enabled** checkbox, using which you can enable the data server and make it ready to use.
- The **Connections** tab contains connection-related information for the data server. The parameter list in this tab varies depending upon the type of connector.

- The **Usage** tab shows a list of names of objects for which the selected connector is used in SAP MII.
- The **Status** tab displays the status of the selected connector, along with the server name, connector type, used connections, available connections, maximum used connections, and maximum waiting time.

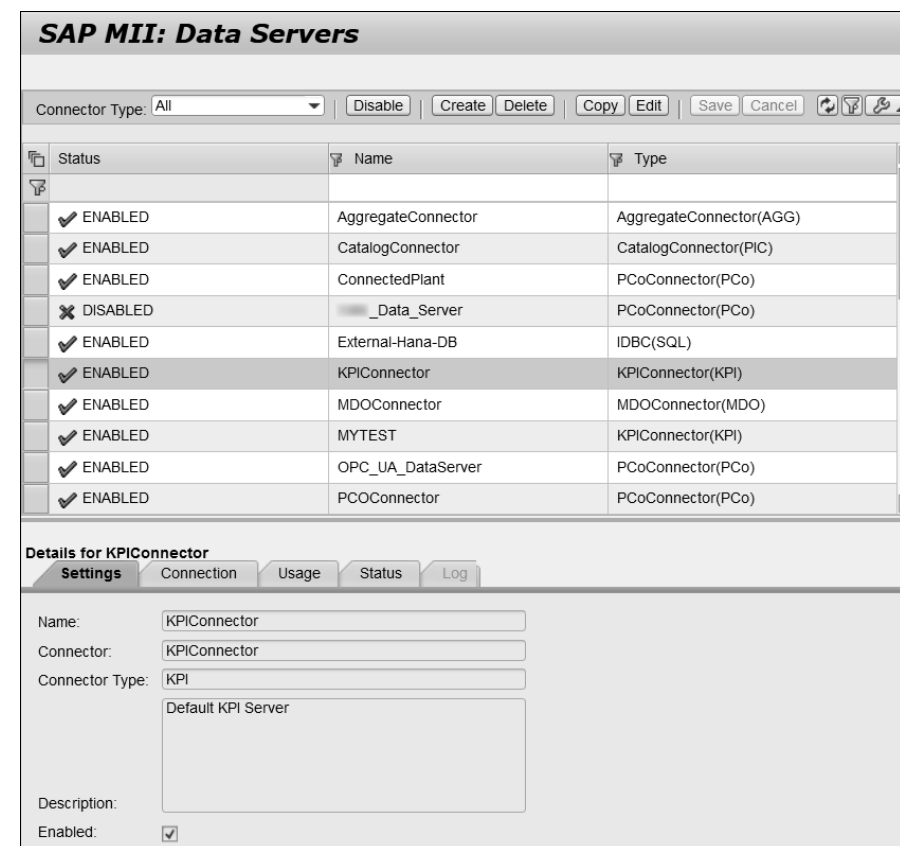


Figure 3.12 Data Server Configuration in SAP MII

**Max. Retry Count**, **Retention Days**, and **Retry Interval** are three connection parameters for the data server that are used for data buffering:

- **Max. Retry Count** indicates the maximum number of times the system will retry execution before stopping.
- The **Retry Interval** value is in milliseconds. It indicates how long the system will wait before retrying.
- **Retention Days** gives the number of days the system will keep the data buffer before removing it.

**Note**

You can get the status of all configured data servers from the following HTTP service:  
<http://<host>:<port>/XMII/Illuminator?Service=SystemInfo&Mode=Status>

**3.2.2 IP21 Connector**

You can connect an SAP MII system with an AspenTech InfoPlus.21 data source using the IP21 connector. Before configuring the IP21 connector, you need to install the IP21 ODBC driver and EasySoft JDBC-ODBC bridge. IP21 connector supports different modes, like the IP21OLEDB connector, discussed in the next section.

**3.2.3 IP21OLEDB Connector**

The IP21OLEDB connector provides connectivity to AspenTech InfoPlus.21, which is a real-time information management system for manufacturing processes. The IP21OLEDB connector uses the SAP MII OLEDB Universal Data Server (UDS) to connect to the AspenTech InfoPlus.21 system.

You can access the data that the InfoPlus.21 suite provides through tag queries, which are created using this data server in the SAP MII workbench. Tag data is time-series data for specific process characteristics, represented by a tag—such as, **Line1Temperature**, **Pump1Flowrate**, and so on. For example, you can retrieve the flow rate of a material or the pressure readings from a production line, which is captured in IP21, in SAP MII by creating a tag query on this data server. You might need this information in various scenarios, such as when you're displaying charts for a process characteristic trend, when doing a quality analysis, when aggregating and updating in SAP ERP or SAP S/4HANA, and so on. The **Mode** settings supported by this connector are **Current**, **HistoryEvent**, **History**, and **Statistics**. It also enables browsing of the IP21 namespace. Because no group hierarchy exists in the IP21 database, the **GroupList** mode does not return any value. In Table 3.9, you can find the general parameters for IP21OLEDB data server configuration.

Property	Description
<b>Message IP</b>	IP address or hostname of the server where the UDS is installed.
<b>Port</b>	The port of the remote machine where the UDS is listening for requests.
<b>TagList</b>	The name of the table that holds a list of all the valid tag names in IP21. Default value is "alltags."

**Table 3.9** General Parameters for IP21OLEDB Data Server Configuration

Property	Description
<b>TagListName</b>	The column name of the <b>TagList</b> that holds the tag names. Default value is "record_name."
<b>TagListDescription</b>	The column name of the <b>TagList</b> that holds the tag description. Default value is "description."
<b>TagInfo</b>	The name of the table from which tag info metadata like description, record type, and range information are fetched for a tag. The default value should be left blank, in which case the name of the tag is used as the table name.
<b>TagInfoName</b>	The column name of the <b>TagInfo</b> that holds the tag names. Default value is "name."
<b>TagInfoDescription</b>	The column name of the <b>TagInfo</b> that holds the tag description. Default value is "name->description."
<b>TagInfoType</b>	The column name of the <b>TagInfo</b> that specifies the IP21 record number. SAP MII interprets #19 and #1649 as discrete data types and #20 and #1640 as analog data types. The rest are interpreted as analog data types. Default value is "definition."
<b>TagInfoMinRange</b>	The column name of the <b>TagInfo</b> that provides EGU range information used for scaling charts using server provided ranges. This field provides the minimum range value. Default value is "name->graphminimum."
<b>TagInfoMaxRange</b>	The column name of the <b>TagInfoTable</b> that provides EGU range information used for scaling charts using server provided ranges. This field provides the maximum range value. Default value is "name->graphmaximum."
<b>Writable</b>	Specifies whether or not you can update current values to a tag using the TagQuery.

**Table 3.9** General Parameters for IP21OLEDB Data Server Configuration (Cont.)

Apart from the general properties just explained, you need to configure the query mode-specific parameters as explained in the following sections. If a specific query mode is not used for a data server, then that mode-specific parameter need not be configured.

**History Mode Parameters**

**History** mode enables you to retrieve historical time-series data in an interpolated manner available in the tag data source. You need to configure the parameters shown in Table 3.10 for **History** mode. All parameters explained in this table are mandatory for using **History** mode.



Property	Description
<b>HistoryTable</b>	The name of the table from where time interpolated history data should be retrieved. Default value should be the name of the history table, which is commonly named "HISTORY."
<b>HistoryName</b>	Specifies the column name in the <b>HistoryTable</b> that represents the tag name. Default value is "NAME."
<b>HistoryTime</b>	Specifies the column name in the <b>HistoryTable</b> that represents the tag value's timestamp. Default value is "TS."
<b>HistoryValue</b>	Specifies the column name in the <b>HistoryTable</b> that represents the tag value at a specific timestamp. Default value is "VALUE."
<b>HistoryQuality</b>	Specifies the column name in the <b>HistoryTable</b> that represents the quality value of the tag data. Default value is "QUALITY."
<b>HistoryPeriod</b>	Specifies the column name in the <b>HistoryTable</b> that sets the data retrieval period. Default value is "PERIOD."

Table 3.10 History Mode Parameters for IP21OLEDB Data Server Configurations

#### HistoryEvent Mode Parameters

The **HistoryEvent** mode enables you to retrieve historical event data available in the tag data source. The **HistoryEvent** mode supports two query methods: normal and compressed.

The *normal* method is invoked when the value of the **Method** field in the query template is blank. The connector then uses the values of the table and column names in the fields of **HistoryEventName**, **HistoryEventTime**, **HistoryEventValue**, and **HistoryEventQuality** in the connector configuration to retrieve the data. The *compressed* method is not available in the method dropdown and has to be entered manually. This method uses the table and column names in the fields of **CompressedHistoryEventName**, **CompressedHistoryEventTime**, **CompressedHistoryEventValue**, and **CompressedHistoryEventQuality** to retrieve the data. These two methods are relevant when you need to maintain and access data with different granularity. The compressed data storing method in the historian does not store all data points but uses different algorithms to reject data that can be interpreted at runtime. You need to configure the parameters in Table 3.11 for **HistoryEvent** mode.

Property	Description
<b>HistoryEventTable</b>	Specifies the name of the table from where history data should be retrieved. Default value is blank, in which case the name of the tag is used as the table name.
<b>HistoryEventName</b>	Specifies the column name from the <b>HistoryEventTable</b> that represents the tag name. Default value is "NAME."
<b>HistoryEventTime</b>	Specifies the column name from the <b>HistoryEventTable</b> that represents the tag value's timestamp. Default value is "IP_TREND_TIME."
<b>HistoryEventValue</b>	Specifies the column name from the <b>HistoryEventTable</b> that represents the tag value. Default value is "IP_TREND_VALUE."
<b>HistoryEventQuality</b>	Specifies the column name in the <b>HistoryEventTable</b> that represents the quality value of the tag data. Default value is "IP_VALUE_QSTATUS."
<b>CompressedHistoryEventTable</b>	Specifies the name of the table from where history data should be retrieved in case the method used in the query is "Compressed." Default value should be left blank, in which case the name of the tag is used as the table name.
<b>CompressedHistoryEvent-Name</b>	Specifies the column name from the <b>CompressedHistoryEventTable</b> that represents the tag name. Default value is "NAME."
<b>CompressedHistoryEventTime</b>	Specifies the column name from the <b>CompressedHistoryEventTable</b> that represents the tag value's timestamp. Default value is "IP_TREND_TIME."
<b>CompressedHistory-EventValue</b>	Specifies the column name from the <b>CompressedHistoryEventTable</b> that represents the tag value. Default value is "IP_TREND_VALUE."
<b>CompressedHistory-EventQuality</b>	Specifies the column name in the <b>CompressedHistoryEventTable</b> that represents the quality value of the tag data. Default value is "IP_VALUE_QSTATUS."

Table 3.11 History Event Mode Parameters for IP21OLEDB Data Server Configurations

#### Current Mode Parameters

**Current** mode enables you to retrieve tag values at the current timestamp. You need to configure the parameters in Table 3.12 for the tag query using **Current** mode.

Property	Description
<b>CurrentTable</b>	The name of the table from where the current value of the tag data should be retrieved. Default value is blank, in which case the name of the tag is used as the table name.
<b>CurrentNameColumn</b>	Specifies the column name in the <b>CurrentTable</b> that represents the tag name. Default value is "NAME."
<b>CurrentValueColumn</b>	Specifies the column name in the <b>CurrentTable</b> that represents the tag value. Default value is "IP_VALUE."
<b>CurrentTime</b>	Specifies the column name in the <b>CurrentTable</b> that represents the timestamp of the selected tag. Default value is "IP_VALUE_TIME."
<b>CurrentQuality</b>	Specifies the column name in the <b>CurrentTable</b> that represents the quality value of the tag data. Default value is "IP_VALUE_QUALITY."

Table 3.12 Current Mode Parameters for IP21OLEDB Data Server Configurations

You need to configure an instance for the IP21 system in the UDS of type OLE DB UDS and point it to the server where the AspenTech InfoPlus.21 system is installed.

### 3.2.4 InSQL Connector

The InSQL connector provides the connectivity to Wonderware IndustrialSQL Server (InSQL). Using this data server, you can create tag queries from the SAP MII workbench to retrieve process time-series data such as machine readings, temperature, and line speed for a production line as current data or historical data for a time range. The InSQL connector is a special type of IDBC connector because it connects to a JDBC-compliant database provided by InSQL, so most of its configuration parameters are the same as IDBC's parameters. Set the **Use OLEDB** property to **True** for InSQL version 8.x and later, and **False** for InSQL version 7.x and earlier. You use **ServerURL** to specify the connection string to the InSQL JDBC-compliant database, and you specify corresponding authentication details in the **UserName** and **Password** properties.

### 3.2.5 InSQLPCo Connector

The InSQLPCo connector is similar to the InSQL connector except for the type PCo that can be used in SAP Plant Connectivity and the PIC query as a source data server.

### 3.2.6 IDBC Connector

The IDBC connector is the most commonly used connector in SAP MII and provides connectivity to legacy databases supporting Java Database Connectivity (JDBC). This connector provides direct connection to any relational database to create a SQL query on its table data. Examples include Microsoft SQL Server, Oracle Database, SAP MaxDB, DB2, SAP HANA database, and so on. You can also use this connector to connect to any application database, such as an MS SQL Server database of an MES. This is useful when you want to retrieve or update data from the composite application to a custom legacy database or any other application database. You need to change the following properties for authentication while creating a new IDBC data server configuration, as shown in Figure 3.13:

- **UserName**
- **Password**
- **ServerURL**

The screenshot displays the 'SAP MII: Data Servers' configuration window. At the top, there are buttons for 'Disable', 'Create', 'Delete', 'Copy', 'Edit', 'Save', and 'Cancel'. Below these is a table listing several IDBC servers, all with a status of 'ENABLED' and a type of 'IDBC(SQL)'. The 'test' server is selected, and its details are shown in the 'Details for test' section. The 'Connection' tab is active, showing fields for Date Prefix, Retention Days (7), Internal Date Format (yyyy-MM-dd HH:mm:ss), Max. Retry Count (5), Pool Max. (100), Retry Interval [ms] (60,000), Timeout (15), Wait Time (30), Date Suffix, Init Command, JDBC Driver (com.microsoft.sqlserver.jdbc.SQLServerDriver), Password (masked), Pool Size (1), Server URL (jdbc:sqlserver://...:1433;databaseName=...), Use Count (256), User Name, and Validation Query (SELECT GETDATE()).

Figure 3.13 IDBC Data Server Configurations

The `ServerURL` parameter specifies the database connection string that is different for different databases. For example, for MS SQL Server it would be as follows:

```
jdbc:sqlserver://<host>:<port>;databaseName=<dbName>
```

In addition, you need to upload and deploy a JDBC driver from the **System Resources • JDBC Drivers** menu. You also need to specify the database driver package name in the `JDBCdriver` property according to the database used (e.g., `com.inet.tds.TdsDriver` for MS SQL Server 8.0 or `com.microsoft.sqlserver.jdbc.SQLServerDriver` for MS SQL Server). Other properties available in the IDBC data server configuration are as follows:

- The **Init Command** field value is used by the connector executed when a connection pool is established.
- The **Validation Query** field is used to check if the database connection is valid. For example, the `SELECT GETDATE()` query is used for Microsoft SQL Server.
- The **Pool Max** field value indicates the maximum number of connections allocated by the pool.
- The **Pool Size** field value specifies the number of connections that will be established during the initialization of the connection pool.
- The **Use Count** field value indicates the number of times a connection is used before it's terminated, releasing the resource.
- The **Wait Time** field value indicates the amount of time the SAP MII system will wait for a connection before the connection is failed.

You can use the default values provided with the new configuration for other required properties.

### 3.2.7 OLAP Connector

The OLAP connector provides connectivity to multidimensional data sources such as online analytical processing (OLAP)-based systems supporting the XML for Analysis (XMLA) specification. Examples of OLAP data sources include SAP BusinessObjects Business Intelligence (SAP BusinessObjects BI) and MS SQL Server Analysis Services. The key elements of an OLAP data source are InfoCubes, multidimensional data structures used for faster execution of complex analytical queries. You can use this data server to create a Multidimensional Expressions (MDX) query over an InfoCube or XMLA data source, the output of which you can use in analysis charts and reports. Some examples might be product yield reports sorted by work centers and by months. The configuration properties in Table 3.13, which are specific to the OLAP data server, are available in the OLAP connector configuration.

Property	Description
<b>Catalog</b>	Specifies the XMLA server or database instance name. For SAP BusinessObjects BI, it specifies the InfoCube name.
<b>ColumnNameAttribute</b>	Specifies the string name used by the XMLA server implementation to retrieve the returned column label. For MS SQL Server Analysis Services, the default value is "sql:field." For SAP BusinessObjects BI, the default value is "name."
<b>DataSourceInfo</b>	Specifies the Microsoft parameter for connecting to a data store (e.g., <code>Provider=MSOLAP; Data Source=local</code> ). Not required for SAP BusinessObjects BI.
<b>DataTypeAttribute</b>	Specifies the string in XMLA used to return the column data types in the response document. Default value is "type."
<b>DecimalDelimiter</b>	Specifies the character used for decimal notation. It can be either a period (.) or a comma (,), as configured in the OLAP system.
<b>IP</b>	Specifies the IP address of the server hosting the OLAP services.
<b>Port</b>	Specifies the listening port of the XMLA service on the OLAP server.
<b>ResultInNamespace</b>	Specifies the Boolean indicator for returning a SOAP response from the XMLA server. Specify <b>True</b> for SAP BusinessObjects BI.
<b>WebService</b>	Specifies the entry point to the XMLA service in the OLAP server. Specify <code>/sap/bw/xml/soap/xmla</code> for SAP BusinessObjects BI and <code>/xmla/msxisapi.dll</code> for MS SQL Server Analysis Services. You need to enable the XMLA web interface in SAP BusinessObjects BI for using the OLAP data server.

Table 3.13 OLAP Data Server Configuration Properties

#### Note

The OLAP data server status doesn't display any number of available or used connections. This is because OLAP is an on-demand connection, not a persistent connection, meaning that the connection is established not by a connection pool, but only when required to execute a query.

### 3.2.8 Open Connector

You can use the open connector to connect to tag-based relational databases to retrieve historical data based on time periods. Using this connector, you can connect

SAP MII to process historian databases, which store process characteristic data in a time-series fashion and provide the following type of tag-based data:

- Current values
- Historical (time-series) values
- Statistical summaries of historical data

In contrast to the previous query modes, the open connector also supports querying tag groups and tag names from the tag data source using the **GroupList** and **TagList** modes. You can create a tag query in SAP MII using this data server to retrieve current or historical process characteristic information such as temperature readings and flow rates. You can use this data server for a custom process historian using any relational database such as Oracle, MS SQL Server, or DB2 from which you can query time-series data. It also acts as a tag data source. The **TagListQuery**, **CurrentQuery**, **HistoryQuery**, and **HistoryEventQuery** parameters should contain the SQL query to select the tag data.

You can use this connector only to query the data from the tag database; you can't write data back to it. Because the open connector uses JDBC data sources, the connection properties such as **JDBC Driver**, **ServerURL**, **UserName**, and **Password** are the same as for the IDBC connector. Apart from these properties, the open connector has certain specific parameters in the data server configuration, as shown in Figure 3.14.

Figure 3.14 Open Connector Data Server Configurations

However, not all these parameters are mandatory; it depends on which query modes of the open connector the data server instance you want to use supports. The following sections explain the parameters required for different query modes.

### GroupList Mode Parameters

**GroupList** mode enables retrieving information about tag groups available in the tag data source. You need to configure the parameters in Table 3.14.

Property	Description
<b>GroupListName</b>	Specifies the column name that returns the tag group name in the <b>GroupList</b> query. Default value is "Name."
<b>GroupListParent</b>	Specifies the column name from the <b>GroupList</b> query that returns the group's parent, if applicable.
<b>GroupListQuery</b>	Specifies the SQL query string that returns the list of tag groups available within the data source. It should include and match the <b>GroupListName</b> and <b>GroupListParent</b> as specified in the corresponding properties, as required.

Table 3.14 GroupList Mode Parameters for Open Connector Data Server Configuration

### TagList Mode Parameters

**TagList** mode enables you to retrieve tag information and metadata from the tag data source. To use this mode, you need to configure the parameters in Table 3.15.

Property	Description
<b>FloatTypeIndicator</b>	Specifies an integer to indicate the tag data type. The floating-point data type is specified as 6, which is a default setting and should not be changed.
<b>IntegerTypeIndicator</b>	Specifies an integer to indicate the tag data type. The integer data type is specified as 5, which is a default setting and should not be changed.
<b>StringTypeIndicator</b>	Specifies an integer to indicate the tag data type. The string data type is specified as 12, which is a default setting and should not be changed.
<b>TagInfoName</b>	Specifies the column from the <b>TagInfoQuery</b> that represents the tag name.
<b>TagInfoDescription</b>	Specifies the column from the <b>TagInfoQuery</b> that represents the tag description.

Table 3.15 TagList Mode Parameters for Open Connector Data Server Configurations

Property	Description
TagInfoMinRange	Specifies the column from the <b>TagInfoQuery</b> that represents the minimum engineering range.
TagInfoMaxRange	Specifies the column from the <b>TagInfoQuery</b> that represents the maximum engineering range.
TagInfoType	Specifies the column from the <b>TagInfoQuery</b> that represents the tag's data type.
TagInfoQuery	Specifies the SQL query string that returns the tag information.
TagListName	Specifies the column from the <b>TagList</b> query that represents the tag name.
TagListDescription	Specifies the column from the <b>TagList</b> query that represents the tag description.
TagListQuery	Specifies the SQL query string that returns the tag list.

Table 3.15 TagList Mode Parameters for Open Connector Data Server Configurations (Cont.)

### History Mode Parameters

**History** mode enables you to retrieve historical time-series data in an interpolated manner available in the tag data source. You need to configure the parameters in Table 3.16 for this mode.

Property	Description
HistoryName	Specifies the column name in the <b>HistoryQuery</b> that represents the tag name.
HistoryTime	Specifies the column name in the <b>HistoryQuery</b> that represents the tag value's timestamp.
HistoryValue	Specifies the column name in the <b>HistoryQuery</b> that represents the tag value at a specific timestamp.
HistoryQuery	Specifies the SQL query string that returns the historical data from the tag data source.

Table 3.16 History Mode Parameters for Open Connector Data Server Configurations

### Current Mode Parameters

**Current** mode enables you to retrieve tag values at the current timestamp. You need to configure the parameters in Table 3.17 for this mode.

Property	Description
CurrentNameColumn	Specifies the column name in the <b>CurrentQuery</b> that represents the tag name.
CurrentValueColumn	Specifies the column name in the <b>CurrentQuery</b> that represents the tag value.
CurrentTime	Specifies the column name in the <b>CurrentQuery</b> that represents the timestamp of the selected tag.
CurrentQuery	Specifies the SQL query string that returns the current tag values from the data source.

Table 3.17 Current Mode Parameters for Open Connector Data Server Configurations

### 3.2.9 OLEDB Connector

OLEDB is a Microsoft API to enable access to data sources using the Microsoft component object model (COM). The SAP MII OLEDB connector provides access to OLEDB-compliant data sources via SAP Plant Connectivity. You can use this data server to connect to MS Access, MS Excel files, OSIsoft PI, and GE Fanuc Proficiency Historian and create SQL queries on its data. This capability is useful when you need to retrieve data from or write data into an Excel file that might be generated by or be sent to another plant system. You need to configure the connection parameters as shown in Table 3.18 to create an OLEDB connector in SAP MII.

Field Name	Purpose
Date Prefix	Prefix used for <b>DateTime</b> database columns.
Date Suffix	Suffix used for <b>DateTime</b> database columns.
Internal Date Format	The date format.
Max. Retry Count	Used for data buffering. Indicates the maximum number of times the system will retry execution before stopping.
Retry Interval [ms]	Used for data buffering. The value is in milliseconds. It indicates the amount of time the system will wait before retrying.
IP	The hostname or the IP address of the source system.
Port	The port of the remote system where the OLEDB is listening for requests.
Query Timeout [sec.]	Timeout value of the query in seconds.
Writable	If checked, provides write access to the tags on the data source.

Table 3.18 Connection Parameters of OLEDB Connector

This is a type of PCoConnector, which needs the OLEDB source system and agent in SAP Plant Connectivity. In SAP Plant Connectivity, you need to create the source using the OLE DB source system and you need to create an agent for same source system. You can configure the agent in SAP MII using PCoConnector and access the data source.

### 3.2.10 Simulator Connector

You can use the simulator connector to connect to a simulated tag data source available in SAP MII. You can use this for demonstrating, testing, or debugging purposes. The simulator connector is configured with an XML file that defines groups, tags, and their values. The XML file can be modified using the **Simulator Services** screen. It supports the following modes:

- Current
- CurrentWrite
- History
- HistoryEvent
- ModelList
- Statistics
- TagList

Using this data server, you can create PCoQueries based on the simulated XML-based tag data source. You can use the default simulator data server available or create new ones as required with different names. In the settings, you can only change the **Description**. You can configure the tags available for a simulator data server using the **Simulator** field (which shows available simulator instance name) under **Simulator Services**.

To change the simulator tags for the default connector or to create the simulator tags for a new simulator data server, open the **Simulator Services** page from the menu **Data Services • Simulator Services**. In the **Simulator Tags** tab, select the simulator data server instance name from the **Simulator Instance** dropdown. The tags for the default simulator are already available in SAP MII, as shown in Figure 3.15. Select an existing tag or create a tag by clicking the **Create** button.

In the **Name** property, specify a unique name for the tag. In the **Type** property, the choices are **Analog**, **Discrete**, and **String**. Use **Analog** for decimal values, such as 192.743; **Discrete** returns 1 or 0; and **String** returns alphanumeric values (e.g., X25, D38, etc.). The **Minimum Value** and **Maximum Value** specify the range of the returned value of the tag. If you mark a tag as **Writable**, then it enables you to write a new tag value using the tag queries. In the **Value** property, specify a formula to return a tag value. You can use time and mathematical expressions to specify the formula. After creating one or more tags, you need to add those to simulator groups from the **Simulator Groups** tab.

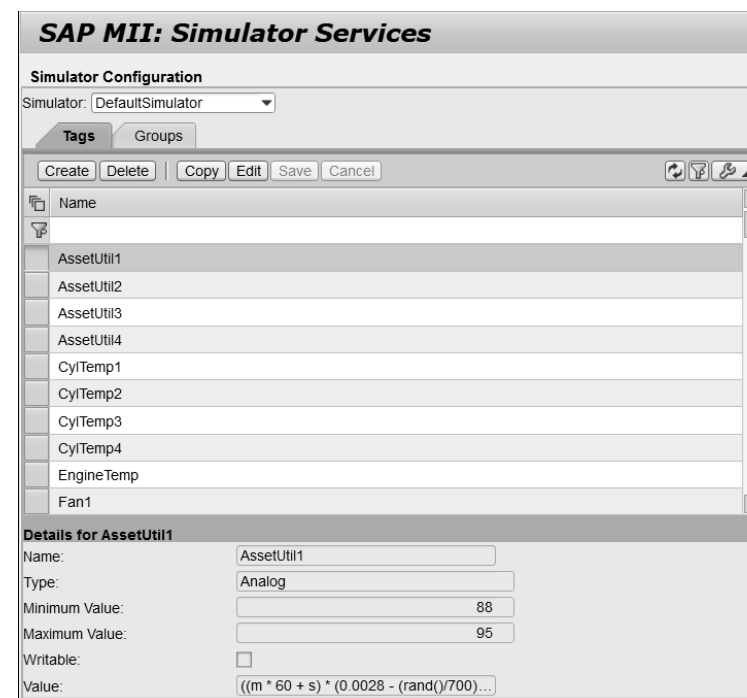


Figure 3.15 Simulator Services

#### Note:

You can use the following characters in the expression for dynamic replacement:

- h: Current hour
- m: Current minute
- s: Current second
- r: Current row number in the result set
- n: Total number of rows in the result set

You can also use mathematical functions such as `rand()`, `sin()`, `cos()`, `log()`, and so on in the expressions for mathematical calculations.

Example expressions are as follows:

- $((m \times 60 + s) \times (0.0097 + (rand() \div 100))) + 815$
- $250 + 250 \times \sin((r \% 60) \times \pi \div 30)$

### 3.2.11 Virtual Servers

In an SAP MII implementation scenario with multiple plants, it's generally advisable to have one instance of SAP MII running per plant, with a separate instance at the corporate level. It's ideally assumed that the plant-level SAP MII server instances will talk and

exchange data among themselves and the central instance, which would mainly serve the purpose of an analytics and reporting dashboard.

You can use virtual server connectors for integration scenarios that involve exchanging data by accessing the data servers of SAP MII server instances residing in different networks and time zones. Data servers not defined in a specific SAP MII server instance's network can still be accessed using virtual server connectors via the SAP MII server on which they're defined as data servers. The data server connector also provides an option for setting an offset to connect to servers in different time zones. This enables the retrieval of corresponding data from the target SAP MII server without changing the existing queries.

There are 10 types of virtual server connectors in SAP MII 15.5: VirtualAlarm, VirtualCatalog, VirtualIDBC, VirtualKPI, VirtualMDO, VirtualOLAP, VirtualPCo, VirtualTAG, VirtualXML, and VirtualXacute. These correspond respectively to their Alarm, Catalog, IDBC, KPI, MDO, OLAP, PCo, TAG, XML, and Xacute connectors. Therefore, using the virtual connectors allows you not only to connect to data server types such as IDBC, OLAP, or any tag-based data source, but also to invoke BLS transactions from another SAP MII installation via the VirtualXacute connector. Using virtual connectors is faster in performance than the usual SOAP or HTTP calls as they use binary mode for data transfer.

To connect to the SAP MII data server using the virtual connector, you need to know the IP or hostname of the server, the username and password of an SAP MII user who has appropriate rights to use the connector, and the name of the data server you're connecting to.

We'll now use an example to demonstrate how interconnectivity can be achieved using the SAP MII virtual server connector. In this example, we'll try to connect to a SQL data server called PlasticResin, which has already been defined in the target SAP MII system using the IDBC connector. The first step is to define a virtual data server in the source SAP MII server to connect with this data server on the target system.

On the **Data Servers Configuration** page, click **Create** and select **VirtualIDBC** as the connector. Then click the **Next** button and select the **Enter Target SAP MII Server Properties Manually** option. Next, click the **Finish** button. On the **Connection** tab, enter the configuration parameters as explained in Table 3.19.

Property	Description
DaysRetention	The number of days the entries of a buffered query should be retained in the buffer.
IP	The IP address/machine name of the remote SAP MII server.
LegacyURL	Check this box to enable support for older SAP MII servers.

Table 3.19 Virtual Server Configuration Properties

Property	Description
MaxRetryCount	Number of times to retry a connection if an attempt fails.
Port	The port to which the remote SAP MII server is listening.
Protocol	Default setting is "http"; however, this can be set to "https" if SSL has been set up.
RemoteLogonName	The login name for the remote SAP MII server (not the remote data server) with proper access rights to the remote data source you want to access.
RemoteLogonPassword	The password for the remote SAP MII server.
RemoteServerName	The name of the data server that you want to access.
RetryInterval	Interval between retry attempts, in milliseconds.
TZ Offset	Describes the time difference between the local and remote SAP MII servers. The time value is measured in minutes.
Destination Name	SAP NetWeaver destination name.
Maintain Session	Used for session connection to avoid multiple logins.
Use Destination Service	Enable when an SAP NetWeaver destination is used and the destination name is provided in the <b>Destination Name</b> field.

Table 3.19 Virtual Server Configuration Properties (Cont.)

After entering the connection parameters, you can enable the server using the **Enable** checkbox in the **Settings** tab. Next, click **Save** to create a new virtual server called **VirtualPlasticResin**. This connects to the PlasticResin data source in the target system, as shown in Figure 3.16.

The last step is to check whether the connection is actually working. Click the **Status** tab to check the server status.

You can now create data queries using the data server, which refers to another data server in a remote SAP MII installation. Launch the SAP MII workbench and create a new SQL query template as explained in Chapter 4. In the data source view, **VirtualPlasticResin** should show up as an available data source. Select **VirtualPlasticResin** as the **Data Source** and set **Mode** to **Query**.

Shift to the **SQL Query Details** configuration page. If a successful connection was made, all the tables from the remote data server should show up under the **Available Tables** list.

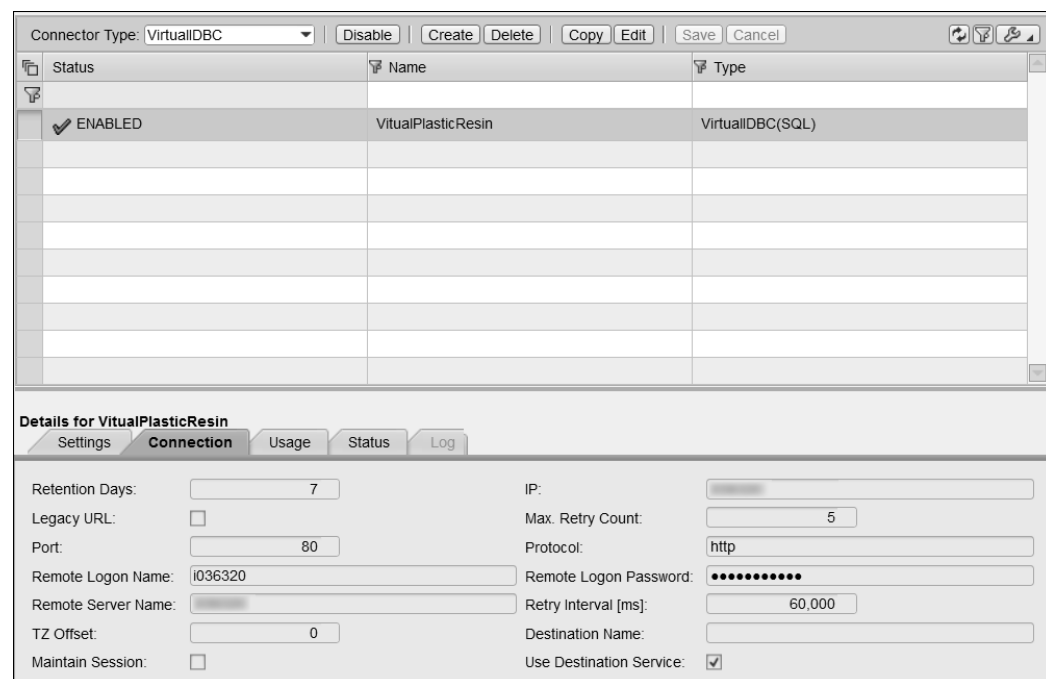


Figure 3.16 Virtual Data Server Configuration

### Troubleshooting

If the connection to the virtual server fails:

- Check whether the username and password you have entered are correct. You should specify the username and password of the remote SAP MII server, not of the remote data source.
- Check whether the target SAP MII server is accessible over the network.
- If you're accessing the remote server through a proxy server, verify whether the proxy server is configured under **System Management • System Properties**.

### 3.2.12 XML Connector

The XML connector is used to connect with an XML-compliant data source to query data from SAP MII. The SAP MII system provides a default XML connector, but you can add other XML connectors. There is nothing to configure on the default XML connector. You need to create XML queries using the XML connector to fetch data from any data source supporting HTTP protocol in XML format. The data query returns an XML document, and the XML data file includes three levels, like the following:

### ■ Rowsets

- Rowset
  - Columns
  - Rows

Rowsets contains attributes like the document creation date, software version, and start date and end date settings used by the query.

### 3.2.13 Xacute Connector

The Xacute connector is an internal system connector for SAP MII that you can use to execute BLS transactions as data queries (called Xacute queries) created using the workbench, which is explained in Chapter 6. You always have a default Xacute data server configuration named XacuteConnector available along with the standard installation of SAP MII. You can use this default configuration for creating and executing Xacute queries. Only the **Description** and **AutoBind** properties are modifiable in the Xacute data server configuration. If the **AutoBind** checkbox is checked, then the SAP MII session variables are automatically mapped to the transaction that has transaction properties with the same name. This is explained in detail with examples in Appendix A, Section A.6.

### 3.2.14 PCo Connector

In SAP MII, the PCo connector is used to connect with an SAP Plant Connectivity instance to perform query functionality. To create the data server of type PCoConnector, you need to go to the SAP MII menu under **Data Servers** and click the **Create** button. The system will open a new page where you can provide the **Server Name** and **Description** and can select **PCoConnector** as the **Connector Type**. Now click the **Next** button and the system will ask you to provide the URL of the SAP Plant Connectivity instance. Either you can enter the URL of the SAP Plant Connectivity management web service manually, or you can browse for it from the System Landscape Directory (SLD). The system will ask you for your user ID and password for the SAP Plant Connectivity management web service interface if required. In the **Connection** tab, you can find the properties described in Table 3.20.

Connection Parameter	Parameter Description
Agent Name	Name of the SAP Plant Connectivity agent created in the SAP Plant Connectivity system.
Agent IP	IP address of the system where the agent is created and SAP Plant Connectivity is installed.

Table 3.20 PCo Connector Connection Properties



Connection Parameter	Parameter Description
Agent Port	Port of the agent configured in SAP Plant Connectivity.
WebSocket Port	You can specify the SAP Plant Connectivity port in the <b>WebSocket Port</b> field; it is used when you use SAP Plant Connectivity as a WebSocket server (e.g., from the UI).
Certificate Keystore	Used from the keystore in SAP NetWeaver Administrator if you want a secure (HTTPS) connection between SAP MII and SAP Plant Connectivity. Used for certificate-based authentication using SSL.
Certificate Name	This names the SSL certificate used for encryption when the SSL connection is used.
Date Prefix	It refers prefix to the date string where date values are compared.
Date Suffix	It refers suffix to the date string where date values are compared.
Internal Date Format	The format used to construct WHERE clause conditions.
Retention Days	The number of days the query remains in SAP MII before being removed.
Max. Retry Count	The number of attempts used to establish the connection.
PCo Mgmt Service URL	The URL of the SAP Plant Connectivity management service ( <i>http://&lt;server&gt;:&lt;port&gt;/PCoManagement?wsdl</i> ).
PCo Mgmt Service User	The username used to access the SAP Plant Connectivity management web service interface.
PCo Mgmt Service Password	The password for the SAP Plant Connectivity management service user.
Query Timeout [sec]	The time in seconds before timeout of a query.
Retry Interval [ms]	The time in milliseconds after which the next connection attempt should be made.
Use SSL	Specifies whether a secured connection will be made between SAP Plant Connectivity and SAP MII via SSL.
Writable	If you enable this checkbox, the system will allow you to run PCoQuery in <b>TagStoreQuery</b> mode so that you can add tag values.

Table 3.20 PCo Connector Connection Properties (Cont.)

**Note**

Before configuring the PCoConnector type data server in SAP MII, you need to install the SAP Plant Connectivity system and perform the following activities in SAP Plant Connectivity:

1. Create a source system in SAP Plant Connectivity.
2. Create an agent using the created source system.
3. Enable the **Run Agent Instance as Executable** checkbox (optional).
4. Configure the query ports.

### 3.2.15 SAP MII and SAP HANA Integration Using SAP HANA Smart Data Integration

The SAP HANA smart data integration adapter helps you get shop floor machine or sensor data into the SAP HANA system virtually without copying data to the SAP HANA system. Using the SAP HANA smart data integration adapter, you can connect to the SAP MII system where the data from the SAP Plant Connectivity queries will be available as virtual tables in SAP HANA. Next, you can use the virtual tables' data for real-time analysis in SAP HANA. To configure the SAP HANA smart data integration adapter and access data from SAP MII, you need to perform the following steps (these steps assume that SAP HANA Studio is available in your local system):

1. Open the **Tools and Archives** page (open *http://<Host>:<Port>/XMII/Menu.jsp* in your browser, then follow menu path **Content Development • Tools and Archives**) in SAP MII to download the SAP MII SDI adapter.
2. After successful download, extract the *MII\_SDI\_Adapter\_1.0.1.jar* file to your local system.
3. Deploy the *MII\_SDI\_Adapter\_1.0.1.jar* file using the SAP HANA Data Provisioning Agent configuration tool.
4. Open SAP HANA Studio and expand the **HANA** system node. To create a remote source system, follow menu path **Provisioning • Remote Sources**, then right click and select **New Remote Source**, as shown in Figure 3.17. Enter the values for the required fields, such as **Source Name** (you can provide any name), **Adapter Name** (select **MII\_SDI\_Adapter**), **Source Location** (select the source agent by which the adapter is deployed), **Server Name** (enter the SAP MII server name), **Port** (provide the SAP MII system port), **SSL Mode** (choose **Enabled** or **Disabled**), **String Column Length** (specify the column length of all the string columns), **User Name**, and **Password** (provide the user name and password for connecting to the SAP MII system).
5. Save the new connection by clicking **Save**.
6. You can view all the SAP MII projects by expanding the **Remote Source System** node.

7. To add the virtual tables, you can select any PCo query from the project, then right-click and select **Add as Virtual Table**. Next you can select the schema and select **Create**. After that, virtual tables will be added to the selected schema.

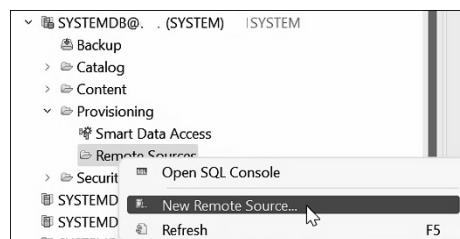


Figure 3.17 SAP MII Integration with SAP HANA Studio

### 3.3 Data Access

The **SAP MII: Data Access** screen, shown in Figure 3.18, enables the administrator to assign SAP MII user roles to the data servers defined using the **Data Servers** screen (explained in Section 3.2) under the **Data Services** menu category. Without the roles assigned to the data server, no data query can be created or executed by the user using that specific data server.

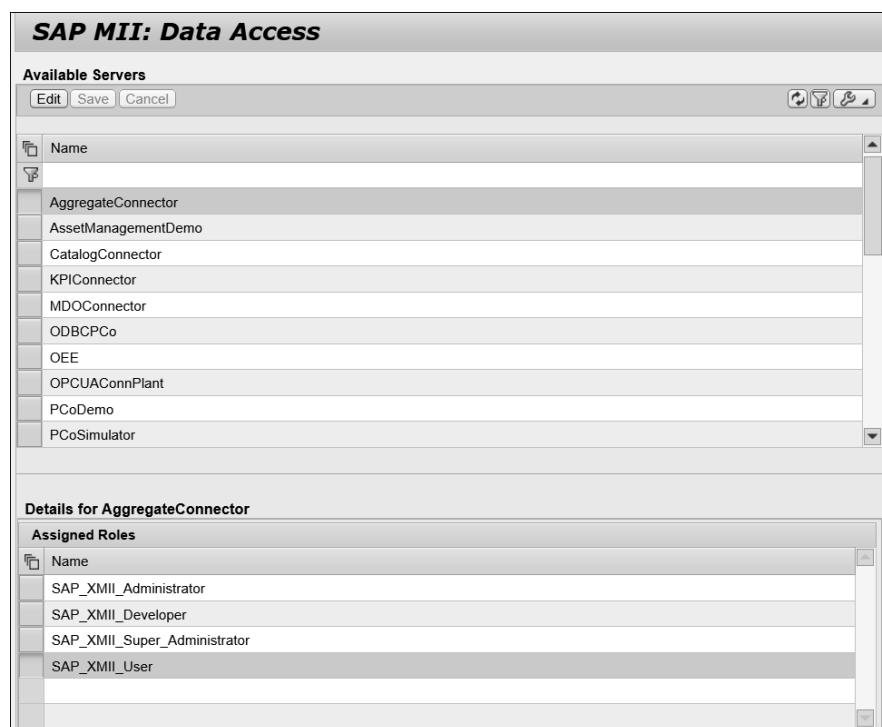


Figure 3.18 Configuring Data Access

To assign roles to a data server, you need to select the data server from the **Available Servers** list and click the **Edit** button. Two tables, **Available Roles** and **Assigned Roles**, will be displayed at the bottom of the **Available Servers** list table. You can search for roles in the **Available Roles** table, then assign the role to the **Assigned Roles** table using the **Add** or **Add All** button. As per your requirements, you can also remove roles from the **Assigned Roles** table by using the **Remove** or **Remove All** button. When the assignment or removal is complete, click the **Save** button.

### 3.4 Message Services: Synchronizing the Shop Floor with the Enterprise

Data captured and generated in enterprise systems (e.g., SAP ERP or SAP S/4HANA), such as production orders, material inventory data, and work center capacity information, might need to be synchronized with the manufacturing shop floor for detailed execution planning and actual execution. Shop floor data synchronization can be done through SAP MII, where shop floor manufacturing-related master data as well as order data are downloaded from SAP ERP or SAP S/4HANA to SAP MII and shop floor manufacturing confirmation data are sent from SAP MII to SAP ERP or SAP S/4HANA. To synchronize data between SAP ERP or SAP S/4HANA and the shop floor, you need to perform required activities that are discussed in the following sections. In Section 3.4.1, you'll learn about synchronizations and SAP MII message services. You'll learn about the required configurations for message listeners in SAP MII in Section 3.4.2 and about message processing rules configuration in SAP MII in Section 3.4.3. Searching and monitoring messages via the message monitor is discussed in Section 3.4.4, and how to clean up the buffered messages from SAP MII is described in Section 3.4.5.

#### 3.4.1 Process Overview

You can synchronize enterprise system data with the manufacturing plant floor systems in two ways:

- In a data pull, the manufacturing integration systems pull the data from the enterprise systems on an as-needed basis.
- In a data push, the data is pushed by the enterprise when required, automatically based on certain events or user actions.

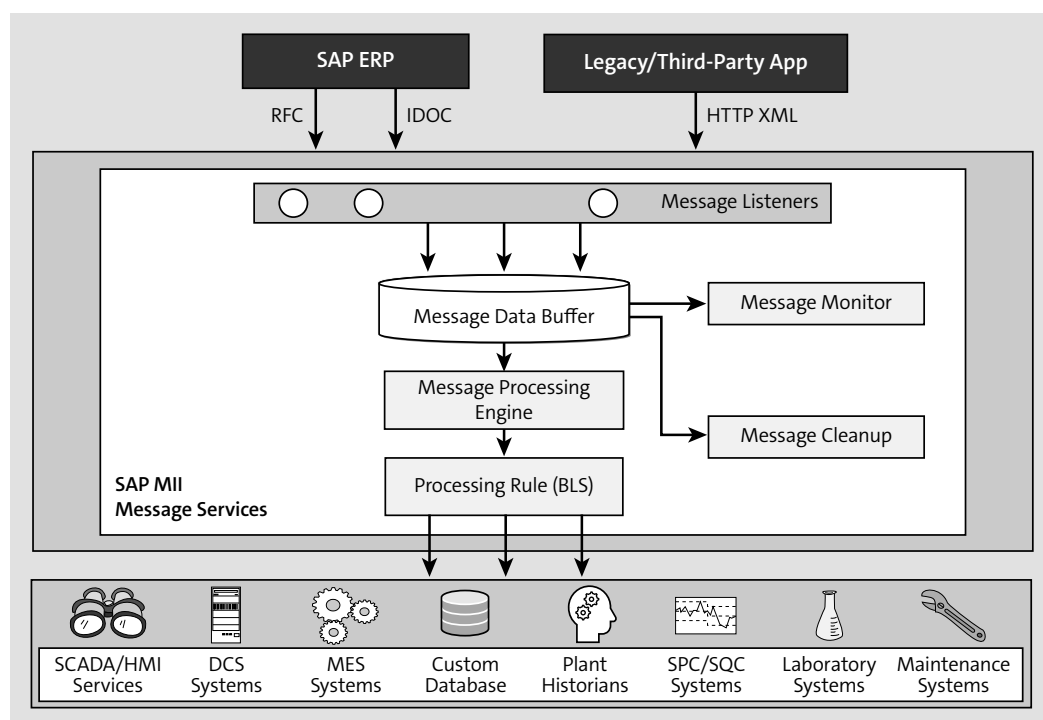
The pull-based method mostly executes RFCs or enterprise services provided by enterprise systems from SAP MII using the SAP Java Connector, SAP Java Resource Adapter, and web service action blocks. The push-based method is often considered necessary because it enables the enterprise system planning and sending data to the plant floor when required. In addition, changes in master data, inventory data, or manufacturing order release can trigger the synchronization of data. The message services module

enables users to receive data sent by the enterprise systems. Message services provides the functionality to synchronize enterprise data, such as data from SAP ERP (or SAP S/4HANA) to SAP MII, by asynchronous data transfer. You can use the message services to receive IDoc and RFC messages from SAP systems and also XML messages by HTTP POST from any enterprise systems. The list of inbuilt SAP MII message listeners is provided in Table 3.21.

Message Listeners	Purpose
XMIIDOC01 to XMIIDOC10	IDoc listener used for IDoc-based communication
XMIIMESSEAGELISTENER	Used for HTTP XML communication
XMIIRFC01 to XMIIRFC10	RFC listener used for RFC asynchronous communication
XMIISRFC01 to XMIISRFC10	Synchronous RFC listener used for synchronous RFC communication

**Table 3.21** SAP MII Message Listeners

The messages received in SAP MII are buffered into the system database, and you can either process them automatically according to predefined rules or categorize them for ad hoc processing. The architecture of message services is shown in Figure 3.19.



**Figure 3.19** SAP MII Message Services

The main functionalities provided by message services in SAP MII are as follows:

- Receive IDoc, RFC, and XML messages by HTTP POST
- Process messages according to processing rule or category
- Monitor buffered messages
- Clean up buffered messages

You can configure message listeners in SAP MII to receive RFC or IDoc messages from enterprise systems asynchronously. The message listeners use SAP Java Connector, and you need to configure different instances of it for each sender system. The message listener for XML messages by HTTP POST is predefined and preconfigured in SAP MII, and you can use it to receive messages from any system which can send XML messages using HTTP POST.

After configuring the message listeners, you can use the processing rule editor to define rules for messages received from different sender systems. A processing rule can either process the message data automatically or assign the message to a category for ad hoc processing. Whenever a message is received by the message listener in SAP MII, it's first saved in the database for buffering and then, if a predefined processing rule exists for it, processed automatically. The messages received and buffered in SAP MII can also be deleted automatically by defining message cleanup rules. The message cleanup rules run periodically for conditionally deleting messages from the database. You can monitor the messages received by the different message listeners from the message monitor and reprocess or delete them individually if required.

#### Example

You can use message services in many scenarios in SAP MII composite applications. One such example is receiving the LOIPRO IDoc for production orders or control recipe data (by RFC) from SAP ERP/SAP S/4HANA as we discussed in Chapter 1. SAP MII buffers the data as an XML message in the database, and you can use a predefined rule or an ad hoc processing rule to parse the message data, extract the relevant information, and populate it in a custom database table. You can then query this table and populate the production operator dashboard developed in SAP MII, from which the user can view the schedule for a work center and subsequently perform order confirmation.

#### 3.4.2 Configuring the Message Listeners

The message listeners in SAP MII are of four types: two listeners for RFC, XMIIRFC[number] adapter for asynchronous and XMIISRFC[number] adapter for synchronous connection; IDoc messages, which you can configure for each sender system; and the XMIIMESSEAGELISTENER message listener, which is a single preconfigured listener for receiving XML messages sent by HTTP POST from any system. Initially, you'll see only a

single preconfiguration, for the message listener named XMIIMESAGELISTENER, as shown in Figure 3.20.



Figure 3.20 HTTP XML Message Listener

This is the preconfigured message listener for receiving XML messages by HTTP POST. You can also edit the existing XMIIMESAGELISTENER and add a schema, XPath for the message name, and XPath for the message UID. After adding the schema, the system will help you get the message name and message UID from there. To send an HTTP XML message to SAP MII, the message needs to be sent to the following URL using HTTP POST as the payload XML:

```
http://<host>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessage-
ListenerServer&NAME=<UniqueMessageName>&IllumLoginName=<username>&Illum-
LoginPassword=<password>&Content-Type=text/xml
```

Replace the `<host>` and `<port>` elements in the URL with the hostname and HTTP port, respectively, of the SAP MII server. Replace the `<UniqueMessageName>` element with the name of the message as specified in the processing rule for the message. If you don't specify a value, the root element name of the XML message is considered the message name, and the processing rule is determined accordingly.

SAP MII also provides quality of services (QoS) functionality where you can specify that messages will be processed exactly once (EO) or exactly once in order (EOIO). The MessageUID is required to process the messages. If you specify EOIO, then the messages will be processed one after another in the queue as those are received. If you specify EO, then messages will be processed in any order inside the queue.

This is the URL for EO:

```
http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessage-
ListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>&Con-
tent-Type=text/xml
```

This is the URL with sequence and message numbers for EOIO:

```
http://<server>:<port>/XMII/Illuminator?service=WSMessageListener&mode=WSMessage-
ListenerServer&NAME=<UniqueMessageName>&MESSAGEUID=<UniqueMessageUID>&SEQUEN-
CENAME=<SequenceName>&MESSAGENUMBER=<MessageNumber>&Content-Type=text/
xml
```

You can configure a maximum of 10 RFC and IDoc message listeners (each) as required, which are provided in SAP NetWeaver Administrator. To configure an SAP MII message listener for IDocs or tRFCs, you need to create an RFC destination. Login to the SAP ERP or SAP S/4HANA system and go to Transaction SM59. Now, you can select **TCP/IP Connections** and click the **Create** button. You need to enter a value for **Destination** (the name of the RFC destination), select **T** as the **Connection Type**, and provide a value for **Description**. Next, press the **Enter** key and move to the **Technical Settings** tab page on the **RFC Destination** screen. Enter the required data as shown in Table 3.22. After entering the required data in the **Technical Settings** tab, you can save the entries.

Field	Description	User Action and Value
<b>Activation Type</b>	Radio buttons	Choose <b>Registered Server Program</b>
<b>Registered Server Program: Program ID</b>	Identification of a registered RFC server program	Enter the same name as the RFC destination for the program ID
<b>Gateway Options: Gateway Host</b>	Gateway host name	Enter the SAP system application server
<b>Gateway Service</b>	Gateway service	Enter "sapgw <SAP system number>"

Table 3.22 Creation of RFC Destination

Now, open `http>>://<Host>:<Port>/nwa` and go to the path **Configuration • Infrastructure • Application Resources** to configure the SAP MII message listener. On the **Application**

**Resource** page select **Resource Adapters** from the **Show** dropdown field. The system will display all available resource adapters in the table. Either you can filter for a specific resource adapter (e.g., XMIIIDOC01) by providing the resource adapter name in the **Resource Name** column or you can select the specific resource adapter from table. Now you can go to the **Properties** tab at bottom of the **Resource Details** area. You need to enter the required values as shown in Table 3.23. After you're finished, click the **Save** button.

Field Name	Description
Program ID	SAP S/4HANA program ID, which is defined in Transaction SM59
MaxReaderThreadCount	Maximum count of listening servers
SAPClient	SAP S/4HANA client
UserName	SAP S/4HANA service user name
Password	Password of SAP S/4HANA service user
Language	Language
ServerName	SAP S/4HANA application server
PortNumber	SAP S/4HANA system number

**Table 3.23** Resource Adapter Configuration

To test the connection, you need to log onto the SAP S/4HANA system and go to Transaction SM59. Now expand **TCP/IP Connections** and select the RFC destination you have created earlier. Next, click the **Display** icon. The system will display the RFC destination, and you can click the **Connection Test** button. A successful connection will be displayed for the connection test event. Apart from the RFC destination configuration and connection test, you also need to perform required ALE configuration on the SAP S/4HANA side to perform the IDoc communication from SAP S/4HANA to SAP MII.

**Note:**

After performing the resource adapter configuration and successful connection testing from the RFC destination on the SAP S/4HANA side, you need to log into SAP MII using URL `http>>://<Host>:<Port>/XMII/Menu.jsp` and the go to menu path **Message Services • Message Listeners**. On the **Message Listeners** page, you need to select configured message listeners (e.g., XMIIIDOC01, which is configured in the **Resource Adapter** settings) and click the **Update** button to update the configuration of the message listener. On the **Status** tab page, you'll see the connection status.

### 3.4.3 Configuring the Processing Rules for Messages

Processing rules in message services are defined to process messages automatically when received from an external sender system. The processing rules can be existing BLS transactions that you can use to process the message data received or to send the message data down to manufacturing plant floor systems. A processing rule is defined for each combination of a message listener and a message name. You can define two types of processing for the messages: transaction-based and category-based.

In transaction-based processing, you need to specify a BLS transaction as the processing rule. In category-based processing, you need only to assign the message to a category (no automatic processing is done). This means that when a specific message is received by the message listener, it's buffered into the database, and the corresponding processing rule is searched. If a BLS transaction is found as the processing rule, then that transaction is executed, passing the XML message data as an input parameter as specified in the processing rule. If a category is found for the processing rule, then the message is buffered into the database and assigned to a category. You can use the message category to logically group messages for ad hoc processing. You can use the message data and other details (e.g., received timestamp, message listener, message name, etc.) in the message services actions in BLS, where the messages can be selected for a specific category as well.

For example, you can use transaction-based processing when you receive production order details from SAP ERP or SAP S/4HANA. SAP MII sends them to the MES for execution, using a BLS as a processing rule. You can use category-based processing where SAP MII doesn't send the messages to any system and buffers the message in the message service data store with a category or tag. You can provide a BLS transaction where the messages are queried based on certain criteria using the message service action blocks (explained in Chapter 6, Section 6.4.8), which an external system can execute as a web service to read the messages received and buffered in SAP MII.

To configure the processing rules, open the **Message Processing Rules** page from the **Message Services** menu category in SAP MII. If no processing rule has been created before, then the message processing rule table doesn't contain any items. Now let's discuss how to configure transaction-based and category-based processing rules. On the **Message Processing Rules** page, click the **Create** button. The system will open a detail page where you can configure the processing rules as described in Table 3.24.

Field Name	Required Configuration
Name	Unique name of processing rule.
Description	Description of processing rule.

**Table 3.24** Message Processing Rules

Field Name	Required Configuration
Message Listener	Select the message listener server from available list to which the rule will be applied.
Message Type	The message type will be selected automatically based on message listener selection. For example, message type <b>IDOC</b> is selected for message listener <b>XMIIDOC01</b> to <b>XMIIDOC10</b> ; message type <b>Web Service</b> is selected for message listener <b>XMIIMESSAGELISTENERS</b> ; and message type <b>RFC</b> is selected for <b>XMIIRFC01</b> to <b>XMIIRFC10</b> and <b>XMIISRFC01</b> to <b>XMIISRFC10</b> .
Message Name	Specify the message names for IDoc, web service, and RFC messages. You can also provide the value * for the message name, which applies to all messages.
Processing Type	Here, you can select the transaction or category processing type. In the <i>transaction</i> processing type, messages are processed immediately through a transaction. In the <i>category</i> processing type, messages are persisted in a persistent queue where you can use the message services actions from the SAP MII workbench to retrieve those messages. If you select the <b>SRFC</b> message type, then you won't have the option to select a category processing type.
Processing Section	The fields in processing sections are dynamically available based on processing type selection from the <b>Processing Type</b> field.
Transaction	This field will be available if you select a transaction processing type. You can browse the transaction created in the SAP MII workbench.
Persist Transaction	You can select an appropriate option here: <b>Always</b> , <b>On Error</b> , or <b>Never</b> .
Log Level	Here you can select <b>Unknown</b> , <b>None</b> , <b>Debug</b> , <b>Info</b> , <b>Warn</b> , <b>Error</b> , or <b>Fatal</b> .
Parameters	The input properties are available as parameters based on selected message listener. These parameters can be mapped to the message fields such as <b>ReceivedMessageXML</b> , <b>MessageUID</b> , and <b>MessageID</b> .
Category	If you select <b>Category</b> for <b>Processing Type</b> , then this field is available in the <b>Processing</b> section. It groups messages and saves them in the buffer for sequential processing. Assigning messages to a category works like a virtual queue.
Description	Holds the description of the category.

Table 3.24 Message Processing Rules (Cont.)

As per the description and your business requirements, you can configure the message processing rules as shown in Figure 3.21.

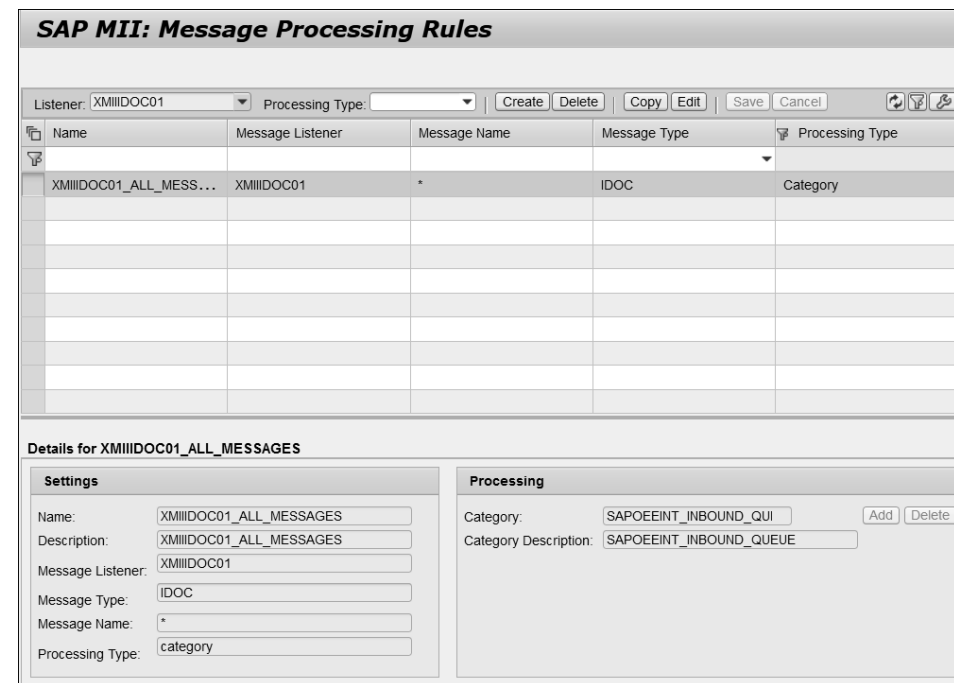


Figure 3.21 Message Processing Rules

### 3.4.4 Using the Message Monitor

You can use the message monitor provided in message services to monitor the status of the messages received. You can also perform additional functions, such as reprocessing, displaying, and deleting messages. The message monitor menu provides the monitoring options for all types of messages. The default selection options in the message monitor display all messages received by any message listener for the last 24 hours (today). You can search messages based on **Listener** selection from the available list, by entering a value (the name of the message you want to find) in the **Find** input field, by selecting an option in the **With Status** dropdown, and by using the **From** field value selection. After specifying search criteria in the required field, you can click the **Go** button to perform your search, as shown in Figure 3.22.

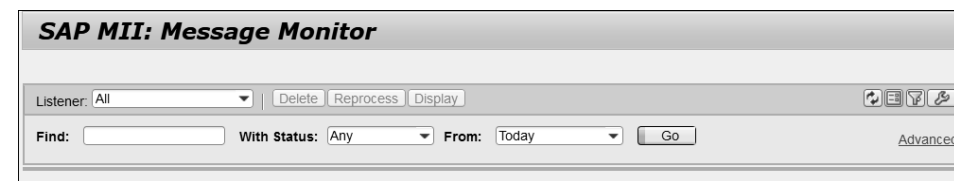


Figure 3.22 Message Monitor Search

In the advanced search criteria, you can specify the details listed in Table 3.25.

Field Name	Usage
Listener	Select the message listener you want to search; select <b>All</b> to search all message listeners
Name	Specify the name of the message you want to search
Category	Select the category of messages you want to search
From	Select a start date for your search criteria
To	Select an end date for your search criteria
Status	Specify the message status you want to search for using multiple checkbox selection, such as <b>Received</b> , <b>Success</b> , <b>Failed</b> , <b>No Rule</b> , <b>Categorized</b> , <b>Running</b> , or <b>Duplicate</b>
Type	Select the message type you want to search for, such as <b>RFC</b> , <b>IDOC</b> , or <b>Web Service</b>

Table 3.25 Advance Search of Message Monitor

After searching, the available messages will be displayed in the message table with columns such as **Received**, **Status**, **Name**, **Type**, **Server**, **Category**, **Message UID**, **Message Number**, and **Processed**, as shown in Figure 3.23. To filter the message display, you can use the following selection criteria:

- **Status**  
Select the specific message status from the dropdown to filter messages on that status. Available options are **Received**, **Success**, **Failed**, **No Ruled**, **Categorized**, **Running** and **Duplicate**.
- **Name**  
Select specific message names (e.g., LOIPRO) to display only those types of messages.
- **Type**  
Select specific message types such as IDoc, RFC, and web service to display only those types of messages.
- **Server**  
Select the specific message listener from the dropdown. Selecting an asterisk signifies all message listeners defined in SAP MII.
- **Category**  
Select message categories created in the processing rules to display only messages assigned to that category. An asterisk signifies all categories or no categories.
- **Message UID**  
You can filter messages based on the unique identification number of the message.

Figure 3.23 Monitor Messages Display

To display a selected message from the table in a new window, click the **Display** button from the **Message Monitor** page. As per business needs, you can also reprocess a message and delete a message from message table using the **Reprocess** and **Delete** buttons.

#### Warning

Deleting messages permanently deletes them from the SAP MII database. They cannot be retrieved again.

In this section, you learned about how to search and monitor messages in SAP MII. Next let's discuss configuration of message cleanup rules.

#### 3.4.5 Configuring and Using the Message Cleanup Rules

Received messages are buffered in the SAP MII database by default, regardless of whether or not any processing rule is defined for them. Messages buffered in SAP MII are not deleted automatically, but you can define message cleanup rules. The message cleanup rules can be based on the message listener, message type, message name, message age, and processing status. The message cleanup rules run periodically at a specific time interval, which you can specify in **System Management • System Properties • Message Cleanup Interval[Hours]**.

To define a new message cleanup rule, open the editor for message cleanup rules (Figure 3.24) under the **Message Services** category. You can specify a unique message cleanup rule **Name** and an optional **Description**. Also select the specific message listener from

the **Server Name** dropdown, or select an asterisk for any message listener. Select the **Message Type** as **IDoc**, **RFC**, or **Web Service**, and specify the **Message Name** as either a specific message name or with an asterisk for any message. In the **Message Older Than** property, specify the time in hours. Messages buffered in SAP MII longer than this specified time are deleted by the cleanup rule if other criteria are satisfied. You can specify the **Processing Status** property as either **All**, which signifies that messages having any of the statuses in the list will be deleted, or any of the usual message processing statuses that are available, except the **Running** status.

Name	Message List...	Message Type	Message Name	Messages Ol...	Processing ...	Status
DeleteLOIPRO	XMIIIDOC01	IDOC	LOIPRO	72	Success	Disabled

**Details for DeleteLOIPRO**

Name: DeleteLOIPRO  
 Description: Cleanup Rule for LOIPRO Messages  
 Message Listener: XMIIIDOC01  
 Message Type: IDOC  
 Message Name: LOIPRO  
 Messages Older Than: 72 Hours  
 Processing Status: Success  
 Enabled:

Figure 3.24 Message Cleanup Rule

You can view one or multiple message cleanup rules in the **Message Cleanup Rules** table, where you can enable or disable and execute the cleanup rules. Using the **Edit** button, you can edit the cleanup rule properties, such as **Rule Description**, **Server Name**, **Message Name**, **Age**, **Processing Status**, and **Rule Activation Status**. You can enable or disable one or more rules by clicking the **Enabled** or **Disabled** buttons. In addition, you can execute the selected rules on an ad hoc basis by clicking the **Run Rule** button. You can configure the message cleanup run interval (**Message Cleanup Interval**) from SAP MII system properties. You can enter a value (in hours) for how often the message cleanup rule will run.

The messages received by the message listeners can also be retrieved and processed by BLS using the message services action blocks, which are explained further in Chapter 6.

### 3.5 Summary

In this chapter, you learned how to configure the data servers and message services that help you to connect SAP MII with plant floor systems and receive messages from enterprise systems.

In the next chapter, you'll learn about how to develop composite applications on SAP MII, work with the SAP MII workbench, and create query templates using the configured data servers.



# Contents

Preface for the Second Edition .....	15
<b>1 SAP MII as an Industry 4.0 Implementation Platform</b> .....	<b>19</b>
<b>1.1 From Traditional Manufacturing to Industry 4.0</b> .....	<b>19</b>
<b>1.2 The Need for a Composition and Integration Platform to Enable Industry 4.0</b> .....	<b>24</b>
<b>1.3 SAP MII as a Manufacturing Integration Platform</b> .....	<b>27</b>
<b>1.4 SAP MII as a Manufacturing Intelligence and Analytics Platform</b> .....	<b>29</b>
<b>1.5 SAP OEE Management as a Manufacturing Performance Management Platform</b> .....	<b>34</b>
<b>1.6 Application Architecture of SAP MII</b> .....	<b>35</b>
<b>1.7 Use Case Scenario</b> .....	<b>37</b>
<b>1.8 Summary</b> .....	<b>38</b>
<b>2 Setting Up SAP MII</b> .....	<b>39</b>
<b>2.1 Software Components and Installation</b> .....	<b>39</b>
<b>2.2 Security and SAP NetWeaver AS Java Administration</b> .....	<b>40</b>
2.2.1 Sizing Determination .....	40
2.2.2 User Management .....	42
2.2.3 Logging and Tracing Configuration .....	44
2.2.4 SAP Java Resource Adapter Configuration .....	46
<b>2.3 SAP Plant Connectivity Installation</b> .....	<b>47</b>
<b>2.4 Postinstallation Configurations of SAP MII</b> .....	<b>53</b>
2.4.1 System Properties in SAP MII .....	53
2.4.2 Active Sessions .....	62
2.4.3 Custom Attributes .....	63
2.4.4 Encryption Configurations .....	65
2.4.5 Personal Data Manager .....	66

2.4.6	Content Management Using NWDI .....	66
2.4.7	Content Transport Using CTS+ .....	77
<b>2.5</b>	<b>Summary .....</b>	<b>82</b>
<b>3</b>	<b>Managing Data from External Systems .....</b>	<b>83</b>
<b>3.1</b>	<b>Using SAP Plant Connectivity .....</b>	<b>83</b>
3.1.1	Source System .....	84
3.1.2	Destination System .....	88
3.1.3	Agent Instance .....	91
3.1.4	Notification .....	100
3.1.5	Managing Notifications from SAP MII .....	105
<b>3.2</b>	<b>Connecting to External Systems Using Data Servers .....</b>	<b>106</b>
3.2.1	Data Server Configuration .....	108
3.2.2	IP21 Connector .....	110
3.2.3	IP21OLEDB Connector .....	110
3.2.4	InSQL Connector .....	114
3.2.5	InSQLPCo Connector .....	114
3.2.6	IDBC Connector .....	115
3.2.7	OLAP Connector .....	116
3.2.8	Open Connector .....	117
3.2.9	OLEDB Connector .....	121
3.2.10	Simulator Connector .....	122
3.2.11	Virtual Servers .....	123
3.2.12	XML Connector .....	126
3.2.13	Xacute Connector .....	127
3.2.14	PCo Connector .....	127
3.2.15	SAP MII and SAP HANA Integration Using SAP HANA Smart Data Integration .....	129
<b>3.3</b>	<b>Data Access .....</b>	<b>130</b>
<b>3.4</b>	<b>Message Services: Synchronizing the Shop Floor with the Enterprise .....</b>	<b>131</b>
3.4.1	Process Overview .....	131
3.4.2	Configuring the Message Listeners .....	133
3.4.3	Configuring the Processing Rules for Messages .....	137
3.4.4	Using the Message Monitor .....	139
3.4.5	Configuring and Using the Message Cleanup Rules .....	141
<b>3.5</b>	<b>Summary .....</b>	<b>143</b>

<b>4</b>	<b>Developing Composite Applications .....</b>	<b>145</b>
<b>4.1</b>	<b>Projects and Configurations .....</b>	<b>145</b>
4.1.1	Projects .....	145
4.1.2	Content Usage .....	146
4.1.3	Configurations .....	147
<b>4.2</b>	<b>Using the SAP MII Workbench .....</b>	<b>148</b>
4.2.1	Layout .....	149
4.2.2	Content Organization and Management .....	151
4.2.3	Important Menu Items and Functionality .....	152
4.2.4	SAP MII Content Security .....	153
<b>4.3</b>	<b>Using Query Templates to Query Data from External Systems .....</b>	<b>155</b>
4.3.1	Data Source Configuration .....	157
4.3.2	General Configuration .....	158
4.3.3	Date Range Configuration .....	162
4.3.4	Parameters Configuration .....	166
4.3.5	Transformation Configuration .....	166
<b>4.4</b>	<b>Types of Query Templates .....</b>	<b>168</b>
4.4.1	OLAPQuery .....	168
4.4.2	SQLQuery .....	170
4.4.3	PCoQuery .....	175
4.4.4	XMLQuery .....	181
<b>4.5</b>	<b>Summary .....</b>	<b>182</b>
<b>5</b>	<b>Manufacturing Data Objects .....</b>	<b>183</b>
<b>5.1</b>	<b>Types of MDOs and Their Usage .....</b>	<b>184</b>
<b>5.2</b>	<b>Using an MDO as an Application Data Store .....</b>	<b>185</b>
<b>5.3</b>	<b>Creating and Configuring MDOs .....</b>	<b>185</b>
<b>5.4</b>	<b>On-Demand MDO .....</b>	<b>191</b>
<b>5.5</b>	<b>Joined MDO .....</b>	<b>191</b>
<b>5.6</b>	<b>MDO Query .....</b>	<b>194</b>
<b>5.7</b>	<b>OData Modeler .....</b>	<b>198</b>
<b>5.8</b>	<b>Summary .....</b>	<b>199</b>

<b>6</b>	<b>Business Logic Service Transactions</b>	201
<b>6.1</b>	<b>Creating a Transaction</b>	201
<b>6.2</b>	<b>Logic, Tracing, and Miscellaneous Functions</b>	209
6.2.1	Logic Action Blocks	209
6.2.2	Miscellaneous Functions Action Blocks	222
6.2.3	Logging Action Blocks	226
<b>6.3</b>	<b>Working with XML</b>	229
6.3.1	SAP MII XML Output Action Blocks	229
6.3.2	XML Functions Action Blocks	236
6.3.3	Reference Documents Action Blocks	256
<b>6.4</b>	<b>Connecting to Enterprise Systems</b>	259
6.4.1	Creating an SAP Server Alias	260
6.4.2	SAP Java Connector Interface Action Blocks	261
6.4.3	SAP Java Resource Adapter Interface Action Blocks	268
6.4.4	XI Action Blocks	271
6.4.5	Web Service Blocks	274
6.4.6	JMS Action Blocks	278
6.4.7	Message Services Action Blocks	281
6.4.8	Manufacturing Dashboards Action Blocks	288
<b>6.5</b>	<b>Managing Plant Data</b>	290
6.5.1	Data Queries Action Blocks	290
6.5.2	SQL Transaction Support Action Blocks	293
6.5.3	Queuing Action Blocks	294
6.5.4	KPIs and Alerts Action Blocks	296
<b>6.6</b>	<b>Charts, Animated Objects, and SQC Analysis</b>	300
6.6.1	Charts Action Blocks	300
6.6.2	Quality Action Blocks	302
6.6.3	Dynamic Graphics Action Blocks	304
<b>6.7</b>	<b>Web, Email, and File Handling</b>	306
6.7.1	Web Action Blocks	306
6.7.2	Email Action Blocks	310
6.7.3	File I/O Action Blocks	312
6.7.4	SFTP Action Blocks	320
<b>6.8</b>	<b>Executing BLS Transactions</b>	321
6.8.1	Xacute Query	321
6.8.2	SOAP Web Service	323
6.8.3	Runner Service	323
<b>6.9</b>	<b>Schedulers</b>	324

<b>6.10</b>	<b>Transaction Manager</b>	327
<b>6.11</b>	<b>Summary</b>	328
<b>7</b>	<b>Visualization Services</b>	329
<b>7.1</b>	<b>i5 Display Templates</b>	329
7.1.1	i5Grid	330
7.1.2	i5Chart	338
7.1.3	i5SPCChart	351
7.1.4	i5ValueHelp	380
<b>7.2</b>	<b>Web Report Development in SAP MII</b>	382
7.2.1	SAPUI5 Application Development using SAP MII	382
7.2.2	Localization in SAPUI5	396
7.2.3	SAP MII Reports	398
7.2.4	Customizing Content for Mobile Devices	400
7.2.5	Localizing Web Content	401
<b>7.3</b>	<b>Self-Service Composition Environment</b>	403
7.3.1	My Dashboards Tab	403
7.3.2	Design Dashboard tab	404
7.3.3	Templates Tab	408
7.3.4	Global Settings	408
<b>7.4</b>	<b>Navigation Services</b>	408
<b>7.5</b>	<b>Summary</b>	411
<b>8</b>	<b>Plant Information Catalog</b>	413
<b>8.1</b>	<b>Configuring the PIC Category and Property Set</b>	414
8.1.1	Category	414
8.1.2	Property Sets	420
<b>8.2</b>	<b>Plant Information Maintenance</b>	426
8.2.1	Groups	427
8.2.2	Tags	431
8.2.3	Other Functions	438
<b>8.3</b>	<b>Linking to an SAP ERP or SAP S/4HANA Source</b>	440
8.3.1	Production Planning	442
8.3.2	Plant Maintenance	442

8.3.3	Assignment from SAP S/4HANA Source .....	443
8.3.4	Sync PIC Using PIC Scheduler .....	445
<b>8.4</b>	<b>Linking to the Shop Floor System .....</b>	<b>447</b>
<b>8.5</b>	<b>Change Management .....</b>	<b>449</b>
8.5.1	Change List .....	449
8.5.2	Approval Lists .....	452
<b>8.6</b>	<b>Accessing PIC through Web Service .....</b>	<b>454</b>
8.6.1	Plant Information Catalog Service .....	454
8.6.2	Plant Information Maintenance Services .....	456
8.6.3	Change Lists Services .....	462
8.6.4	Property Set Services .....	463
8.6.5	Accessing Plant Information Web Services from SAP MII Transactions .....	464
<b>8.7</b>	<b>Catalog Query .....</b>	<b>466</b>
<b>8.8</b>	<b>Using PIC for Plant Data Collection .....</b>	<b>468</b>
<b>8.9</b>	<b>User Authorization in Catalog Services .....</b>	<b>469</b>
<b>8.10</b>	<b>Summary .....</b>	<b>473</b>
<b>9</b>	<b>Energy Monitoring and Analysis .....</b>	<b>475</b>
<b>9.1</b>	<b>Production Events .....</b>	<b>475</b>
9.1.1	Production Event Templates Overview .....	476
9.1.2	Create Production Event Template .....	477
9.1.3	Create Production Event .....	479
9.1.4	Configuration of Production Event Type Scheduler .....	481
<b>9.2</b>	<b>Key Performance Indicators .....</b>	<b>481</b>
9.2.1	Creating a KPI .....	482
9.2.2	Creating a KPI Query .....	487
9.2.3	KPI Monitor .....	490
9.2.4	KPI Scheduler .....	493
<b>9.3</b>	<b>Energy Tariff .....</b>	<b>495</b>
<b>9.4</b>	<b>Configuring Shift Patterns .....</b>	<b>497</b>
<b>9.5</b>	<b>Configuring Plant Information Catalog .....</b>	<b>499</b>
<b>9.6</b>	<b>PIC Analysis Query Report .....</b>	<b>501</b>
<b>9.7</b>	<b>Summary .....</b>	<b>504</b>

<b>10</b>	<b>Overall Equipment Effectiveness .....</b>	<b>505</b>
<b>10.1</b>	<b>SAP OEE Management Overview .....</b>	<b>506</b>
10.1.1	Component Architecture .....	506
10.1.2	Formula and Calculations .....	507
10.1.3	User Role and Authorization .....	509
10.1.4	Data Model .....	511
<b>10.2</b>	<b>Post-Installation Configuration of SAP OEE Management .....</b>	<b>513</b>
10.2.1	SAP S/4HANA Customization for SAP OEE Management .....	514
10.2.2	SAP S/4HANA Master Data Setup .....	529
10.2.3	ALE Setup .....	542
10.2.4	Central Technical Configuration Wizard .....	550
<b>10.3</b>	<b>ERP Shop Floor Integration for Worker UI .....</b>	<b>561</b>
10.3.1	Global Configurations .....	561
10.3.2	Supported Plants .....	562
10.3.3	Workflow Configurations .....	563
10.3.4	Transfer Configuration and Master Data and Related Workflows .....	568
10.3.5	Queue Monitor .....	571
<b>10.4</b>	<b>General Configuration .....</b>	<b>572</b>
10.4.1	User Group Assignment .....	573
10.4.2	Customization Configuration .....	573
10.4.3	User Group Dashboard Assignment .....	580
10.4.4	Reason Code Configuration .....	581
10.4.5	Extension Configuration .....	582
<b>10.5</b>	<b>Activity Configuration .....</b>	<b>583</b>
10.5.1	Activity Description and Options .....	584
10.5.2	Activities .....	585
10.5.3	Managing Activity .....	589
<b>10.6</b>	<b>Dashboard Configuration .....</b>	<b>590</b>
10.6.1	Dashboard Details .....	590
10.6.2	Dashboard Description .....	591
10.6.3	Dashboard Buttons .....	592
<b>10.7</b>	<b>Modify Order Status Description .....</b>	<b>594</b>
<b>10.8</b>	<b>Data Upload .....</b>	<b>596</b>
10.8.1	Downtimes .....	597
10.8.2	Reason Code Assignment to Machine .....	597
10.8.3	Plant Information Catalog .....	598
<b>10.9</b>	<b>Scheduled Downtime Maintenance .....</b>	<b>598</b>

<b>10.10 Order Dispatch</b> .....	599
<b>10.11 Display SAP S/4HANA Master Data</b> .....	600
<b>10.12 Audit Log</b> .....	601
<b>10.13 SAP OEE Management Dashboard</b> .....	602
10.13.1 Operator Dashboard .....	602
10.13.2 Plant Monitor Dashboard .....	622
10.13.3 Line Monitor Dashboard .....	622
<b>10.14 Application Launchpad</b> .....	623
10.14.1 Setup and Configuration .....	624
10.14.2 Goods Issue .....	625
10.14.3 Goods Receipt .....	627
<b>10.15 Custom Enhancements</b> .....	629
10.15.1 Custom Dashboard .....	630
10.15.2 Custom Extension Development and Configuration .....	633
10.15.3 SAP_OEE_Integration Action Blocks .....	635
<b>10.16 Reporting and Analytics</b> .....	637
10.16.1 Plant-Level Reporting .....	637
10.16.2 Analytics Using SAP HANA .....	639
10.16.3 Analytics Using SAP Digital Manufacturing Cloud .....	640
<b>10.17 Summary</b> .....	642
 <b>Appendices</b> .....	 643
<b>A Advanced Techniques for SAP MII Composite Application Development</b> ...	643
<b>B The Authors</b> .....	689
 Index .....	 691

# Index

.irpt extension .....	398	Annotations (Cont.)	
/sap/bw/xml/soap/xmla .....	117	<i>DefinesAccompanimentData</i> .....	647
/xmla/msxisapi.dll .....	117	<input/> .....	646
		<i>outputs</i> .....	646
<b>A</b>		Append XML .....	659
Accessibility .....	666	Application architecture .....	35
Accumulator .....	246	Application configuration .....	576
Action blocks .....	33, 203, 205	Application data store .....	185
<i>custom</i> .....	635, 643	Application ID .....	388
<i>deployment</i> .....	648	Application launchpad .....	623
Active sessions .....	62	<i>activites</i> .....	625
Activities .....	583	<i>setup</i> .....	624
<i>custom UI</i> .....	632	Application Link Enabling (ALE) .....	542
<i>description</i> .....	584	Approval lists .....	452
<i>manage</i> .....	589	<i>display details</i> .....	453
<i>type extension</i> .....	588	AspenTech InfoPlus.21 .....	107, 110
<i>types</i> .....	585	Asset hierarchy .....	431
Admin service .....	676	Assignment action block .....	210
Adobe Lifecycle Designer .....	318	Asynchronous call .....	218
ADS printing action block .....	318	Asynchronous communication .....	132, 271
Advanced search criteria .....	140	Attachment .....	311
Agent instance .....	91	AttachmentMimeType .....	310
<i>host</i> .....	91	Attribute columns .....	354
<i>log</i> .....	92	Audit log .....	601
<i>messages for bundling</i> .....	99	Authentication .....	79
<i>notification processing</i> .....	96	Authorizations .....	469
<i>queued messages</i> .....	99	Autobind .....	127, 267, 664
<i>servers</i> .....	93	Autocommit .....	269
<i>subscription items</i> .....	96	Availability .....	507
<i>tag query</i> .....	95		
Aggregate statistics action block .....	236	<b>B</b>	
Alarms .....	362	B2MML .....	256
<i>conditions</i> .....	363	BAPI .....	261, 657
ALE/IDoc .....	26, 28	<i>execute</i> .....	266
Alert service .....	681	Bar chart .....	346
Alerts .....	289, 296, 486	Base KPI .....	484
<i>category</i> .....	289	Batch characteristics .....	190
<i>severity</i> .....	298	BLS engine .....	33
Alias .....	238	BLS transactions .....	28, 33, 127, 201
<i>name</i> .....	239	<i>create</i> .....	202
Analytics .....	26, 29	<i>design time</i> .....	204
Animated dynamic visualization .....	30	<i>execute</i> .....	321
Animated graphics .....	29	<i>optimized</i> .....	651
Animation renderer action block .....	305	<i>properties</i> .....	203
Annotations .....	645	<i>schedule</i> .....	324
<i>action</i> .....	645	<i>terminate</i> .....	216
<i>ConfigurationDialog</i> .....	646	Bottleneck machine .....	538

- Break condition ..... 217
  - Break property ..... 213
  - Bubble chart ..... 350
  - Buffered messages ..... 28
  - Buffering ..... 133, 158
  - BufferListDoc property ..... 282
  - Build-time dependency ..... 69
  - Business logic services (BLS) ..... 27, 29, 201
- C**
- C chart ..... 373
  - Calculated columns action block ..... 237
  - Calculation action block ..... 223
  - Catalog ..... 117
  - Catalog objects ..... 151
  - Catalog query ..... 466
    - details* ..... 467
  - Catalog schema ..... 440
  - Catalog services ..... 27, 28
    - authorizations* ..... 469
  - CatalogQuery ..... 501
  - Catch action block ..... 222
  - Categories ..... 137, 414
    - create* ..... 418
    - general parameters* ..... 415
    - icon* ..... 418
    - list* ..... 415
    - localization* ..... 416
    - translation* ..... 419
    - usage* ..... 417
  - Central Technical Configuration (CTC) ..... 511, 550
    - validation* ..... 560
  - Central User Administration (CUA) ..... 43
  - Change and Transport System (CTS+) ..... 36, 39, 77
    - NWDI CMS integration* ..... 80
  - Change list ..... 449
    - details* ..... 451
    - filter* ..... 450
    - log* ..... 451
    - notes* ..... 452
  - Change List Service API ..... 462
  - Change management ..... 449
  - Change Management Service (CMS) . 66, 70, 73
    - functional unit* ..... 69
    - track* ..... 81
  - Chart action block ..... 300, 301
  - Chart servlet ..... 399
  - Charts and visualization controls ..... 30
  - Child Document Source ..... 243
  - Classes ..... 648
  - Cleanup rules ..... 28
  - CM Services ..... 73
  - Color context ..... 334
  - Column action block ..... 233
  - Column aliasing action block ..... 238
  - Column lights grid ..... 333, 336
  - Column map ..... 292
  - Column map action block ..... 292
  - Column stripper action block ..... 239
  - Column to Alias property ..... 239
  - ColumnNameAttribute ..... 117
  - Columns ..... 438
  - Comma-separated values ..... 326
  - Commit ..... 261, 270
    - transaction* ..... 265
  - Communication method ..... 271
  - Comparative charts ..... 30
  - Component build service ..... 72
  - Component controller ..... 393
  - Components list ..... 626
  - Composite applications ..... 26, 145
  - Composite KPI ..... 484
  - Composition capability ..... 28
  - Composition platform ..... 26
  - CompressedHistoryEventName ..... 112
  - CompressedHistoryEventNameColumn ..... 113
  - CompressedHistoryEventQuality ..... 112
  - CompressedHistoryEventQualityColumn ..... 113
  - CompressedHistoryEventTable ..... 113
  - CompressedHistoryEventTime ..... 112
  - CompressedHistoryEventTimeColumn ..... 113
  - CompressedHistoryEventValue ..... 112
  - CompressedHistoryEventValueColumn ..... 113
  - Conditional action block ..... 211
    - legs* ..... 212
  - Conditional expression ..... 212
  - Connection factory ..... 46
  - Connection session ..... 269
  - Connection status ..... 108
  - Connector types ..... 106
  - Connectors ..... 106
  - Content and version management ..... 36
  - Content catalog ..... 307
  - Content management ..... 66
  - Content transport ..... 77
  - Content usage ..... 146
  - ContentList ..... 676
  - ContentType property ..... 309
  - Context properties ..... 296
  - Context-sensitive configuration options ..... 150
  - Control chart ..... 30

- Control limits ..... 361
  - Counter ..... 247
  - Create directory action block ..... 314
  - Create ZIP archive action block ..... 315
  - Created\_By ..... 666
  - Credential alias ..... 273, 276, 278
  - Credential store ..... 494
  - cron pattern ..... 189, 325, 326
  - Crosstab action block ..... 240
  - CSRF token ..... 277
  - CubeList ..... 168
  - Current mode ..... 113, 120
  - CurrentItem property ..... 213, 219, 220
  - CurrentNameColumn ..... 114, 121
  - CurrentProfile ..... 676, 685
  - CurrentQualityColumn ..... 114
  - CurrentQuery ..... 118, 121
  - CurrentTable ..... 114
  - CurrentTimeColumn ..... 114, 121
  - CurrentValueColumn ..... 114, 121
  - Custom attributes ..... 63
    - parameters* ..... 63
  - Custom business logic ..... 26
  - Custom chart ..... 348
  - Custom enhancements ..... 629
  - Custom UI ..... 630
  - Customer distribution model ..... 549
  - Customer executive ..... 26
- D**
- Dashboards ..... 30, 590
    - assign* ..... 580
    - buttons* ..... 592
    - custom* ..... 630
    - custom button* ..... 632
    - description* ..... 591
    - details* ..... 590
  - Data access ..... 130
  - Data buffering ..... 160
  - Data collection elements ..... 518, 519
    - configure* ..... 518, 520
  - Data connectors ..... 27
  - Data element types ..... 514
    - configure* ..... 515
  - Data elements ..... 516
    - configure* ..... 517
  - Data integration ..... 33
  - Data item action block ..... 234
  - Data provider ..... 186, 484
  - Data pull ..... 131
  - Data push ..... 131
  - Data queries ..... 27, 125, 157
  - Data queries action block ..... 290
  - Data reporting ..... 330
  - Data series details ..... 348
  - Data server configuration ..... 108
  - Data server services ..... 675
  - Data servers ..... 106
    - configuration* ..... 124
  - Data services ..... 27, 130
  - Data source ..... 181
  - Data types ..... 202
  - Data upload ..... 596
  - DataSourceInfo ..... 117
  - DataTypeAttribute ..... 117
  - DataValue property ..... 224
  - Date format ..... 330
  - Date ranges ..... 162
  - db:// ..... 307
  - DB2 ..... 115
  - DBInit ..... 676
  - Debug ..... 226
  - Debug messages ..... 226
  - Debug-level logging ..... 291
  - DecimalDelimiter ..... 117
  - Decision support ..... 26
  - Delete alert action block ..... 299
  - Delete file action block ..... 314
  - Delete message action block ..... 285
  - Deleting messages ..... 139
  - Delimiter ..... 249
  - Demonstrating ..... 122
  - Design-time repository ..... 72
  - Destination system ..... 88
    - notifications* ..... 103
  - Destinations ..... 102
  - Development component ..... 68
  - Development infrastructure ..... 76
  - Development track ..... 74
  - DimensionList ..... 168
  - Dimensions ..... 485
  - Disabled ..... 142
  - Discrete data type ..... 415
  - Display templates ..... 30, 329
    - properties* ..... 329
  - Distinct value extractor action block ..... 241
  - Distributed control system (DCS) ..... 24, 26, 27, 288
  - Document action block ..... 231
  - Document object ..... 232
  - Document Object property ..... 233
  - Document source ..... 225, 237, 238
  - Donut chart ..... 350

Downtimes ..... 597  
 Dynamic analysis ..... 30  
 Dynamic background ..... 335  
 Dynamic data ..... 399  
 Dynamic data exchange ..... 668  
 Dynamic graphics action block ..... 304  
 Dynamic link ..... 207  
 Dynamic queries ..... 655  
 Dynamic transaction call action block ..... 218

**E**

Easysoft JDBC/ODBC driver ..... 107  
 Editor for message cleanup rules ..... 141  
 EJB ..... 287, 288  
 EJB callback action block ..... 287  
 Email action block ..... 310  
 Embedded lights grid ..... 337  
 Enabled ..... 142  
 Encoded image ..... 300  
 Encryption configuration ..... 65  
 Encryption method ..... 65  
 Energy monitoring ..... 475, 482  
 Energy monitoring framework ..... 32  
 Energy tariff ..... 495  
 Enterprise business systems ..... 259  
 Enterprise service action block ..... 276  
 Environmental variable ..... 267  
 EPM\_KPI\_DATA\_INSERT ..... 289  
 Error message ..... 235  
 Evaluate ..... 206  
 Event logger action nblock ..... 229  
 Event type ..... 229  
 EWMA chart ..... 368  
 EWMA-RANGE chart ..... 368  
 EWMA-SDEV chart ..... 369  
 Exactly once in order (EOIO) ..... 135  
 Exception enabler action block ..... 221  
 Exception message ..... 222  
 Expression editor ..... 206, 238  
 Extension configuration ..... 582  
 External systems ..... 83  
   *connecting* ..... 106

**F**

Factory calendar ..... 532  
   *assign* ..... 532  
 Fatal error action block ..... 235  
 FatalError ..... 235  
 File I/O action block ..... 312  
 File Path property ..... 227

Filter expressions ..... 657  
 Filtering data ..... 656  
 Filters ..... 242  
 Fixed query ..... 174  
 Flat file parser action block ..... 315, 316  
 FloatTypeIndicator ..... 119  
 For next loop action block ..... 213  
 Forward alert action block ..... 300  
 Fragments ..... 395  
 FromAddress ..... 310  
 FTP create directory action block ..... 315  
 FTP delete file action block ..... 315  
 FTP get file list action block ..... 315  
 FTP input action block ..... 314  
 FTP output action block ..... 315  
 FTP server ..... 314, 320  
 FullName ..... 666  
 FullProfile ..... 676  
 Functional location group ..... 445  
 Functional units ..... 67  
 Functions ..... 206

**G**

Gauge chart ..... 347  
 Generation of alerts ..... 33  
 Generic sort filter action block ..... 242  
 Get alert details action block ..... 300  
 Get alerts action block ..... 299  
 Get file list action block ..... 313  
 getDisplayTemplate() ..... 414  
 getProperty(PROPERTYNAME) ..... 415  
 getQueryTemplate() ..... 414  
 Global hierarchy ..... 534  
 Global hierarchy template ..... 524  
   *node type* ..... 525  
   *relationship* ..... 526  
 Global reason code ..... 528  
   *level* ..... 529  
 Globalization ..... 19  
 Goods Issue app ..... 625  
 Goods Receipt app ..... 627  
 Grid ..... 336  
 Grid type ..... 331  
 Group bar chart ..... 346  
 GroupList ..... 118  
 GroupList mode ..... 119  
 GroupListNameColumn ..... 119  
 GroupListParentColumn ..... 119  
 GroupListQuery ..... 119  
 Groups ..... 427  
   *custom property* ..... 429

Groups (Cont.)  
   *general attributes* ..... 427  
   *logs* ..... 429  
   *manage admin* ..... 430  
   *node authorization* ..... 430  
   *property sets* ..... 428  
   *usage* ..... 429

**H**

Hierarchy node classification ..... 524  
 Hierarchy templates ..... 524  
 HISTOGRAM chart ..... 371  
 Historian ..... 26, 27  
 History ..... 679  
 History mode ..... 111, 120, 436  
 HistoryEvent mode ..... 112  
 HistoryEventNameColumn ..... 112, 113  
 HistoryEventQualityColumn ..... 113  
 HistoryEventQuery ..... 118  
 HistoryEventTable ..... 113  
 HistoryEventTimeColumn ..... 112, 113  
 HistoryEventValueColumn ..... 112, 113  
 HistoryNameColumn ..... 112, 120  
 HistoryPeriodColumn ..... 112  
 HistoryQualityColumn ..... 112  
 HistoryQuery ..... 118, 120  
 HistoryTable ..... 112  
 HistoryTimeColumn ..... 112, 120  
 HistoryValueColumn ..... 112, 120  
 Horizontal bar chart ..... 348  
 Horizontal group bar chart ..... 349  
 HostInfo ..... 685  
 HTML loader action block ..... 307  
 HTML saver action block ..... 307  
 HTTP authentication ..... 307  
 HTTP post action block ..... 308  
 HTTP services ..... 33, 674  
 HTTP XI action block ..... 271  
 HTTP XML ..... 132  
 HTTP XML messages ..... 28

**I**

i5 display templates ..... 330  
 i5Chart ..... 386, 388  
   *chart area configuration* ..... 339  
   *chart background color* ..... 339  
   *chart border color* ..... 339  
   *chart type* ..... 339  
   *configuration* ..... 339  
   *data mapping* ..... 341

i5Chart (Cont.)  
   *data series configuration* ..... 344  
   *horizontal grid line count* ..... 339  
   *label columns* ..... 341  
   *legend* ..... 342  
   *marker size* ..... 339  
   *marker style* ..... 340  
   *number of vertical grid lines* ..... 339  
   *refresh page* ..... 344  
   *specification limits* ..... 344  
   *subtypes* ..... 345  
   *title configuration* ..... 341  
   *value columns* ..... 341  
   *X-axis* ..... 343  
   *Y-axis* ..... 344  
 i5Display ..... 668  
 i5Grid ..... 330  
   *allow column resizing* ..... 333  
   *auto resize columns* ..... 333  
   *color context* ..... 334  
   *columns* ..... 332  
   *configuration tabs* ..... 331  
   *freeze column* ..... 333  
   *grid area* ..... 331  
   *header configuration* ..... 333  
   *layout* ..... 331  
   *row heading configuration* ..... 333  
   *row headings* ..... 333  
   *selection* ..... 333  
   *subtypes* ..... 336  
   *UI behavior configuration* ..... 333  
 i5SPCChart ..... 351  
   *alarms configuration* ..... 362  
   *chart area* ..... 352  
   *configuration* ..... 352  
   *context menu behavior configuration* ..... 363  
   *data mapping* ..... 354  
   *formatting* ..... 354  
   *legend configuration* ..... 356  
   *lower chart area configuration* ..... 362  
   *lower chart spec. limits* ..... 361  
   *parameters configuration* ..... 355  
   *sample size column* ..... 354  
   *subtypes* ..... 364  
   *time stamp column* ..... 354  
   *title configuration* ..... 354  
   *upper chart area* ..... 359  
   *upper chart area configuration* ..... 360  
   *upper chart axes configuration* ..... 359  
   *upper chart control limits* ..... 361  
   *upper chart spec. limits* ..... 357  
   *X-axis configuration* ..... 356



- iSPCChart (Cont.)
    - Y-axis configuration* ..... 357
  - i5ValueHelp ..... 380
    - configuration* ..... 381
    - general configuration* ..... 381
    - search results* ..... 381
  - iChart ..... 300, 338
  - iCommand applet ..... 217, 235
  - IDBC connector ..... 115
  - IDoc ..... 132, 281, 568
    - integration scenario* ..... 41
  - illuminator ..... 399
  - illuminator services ..... 182, 674
  - illuminatorDocument ..... 232
  - illumLoginName ..... 666
  - illumLoginRoles ..... 666
  - Image combiner action block ..... 305
  - Image creator action block ..... 306
  - Image format ..... 305
  - Image loader action block ..... 307
  - Image saver action block ..... 301, 307
  - Image scaler action block ..... 306
  - IMAP4 ..... 312
  - index.html ..... 395
  - Indexing services ..... 677
  - INDIVIDUALS chart ..... 375
  - INDIVIDUALS-MR chart ..... 377
  - INDIVIDUALS-MR-SHORT-RUN chart ..... 378
  - INDIVIDUALS-SHORT-RUN chart ..... 377
  - Industry 4.0 ..... 19, 21
    - enabling* ..... 24
  - Industry4.Now ..... 23
  - InfoCube ..... 116
  - InfoPlus.21 suite ..... 107
  - Information message ..... 235
  - InputDocument property ..... 271
  - InputValue property ..... 215
  - InputXSL property ..... 249
  - InSQL ..... 114
  - InSQL connector ..... 114
  - Installation ..... 39
  - IntegerTypeIndicator ..... 119
  - Integral ..... 247
  - Integration ..... 578
  - Integration capability ..... 27
  - Intelligence system ..... 25
  - Intelligent visualization ..... 29
  - Intercept ..... 225
  - Interface name ..... 271
  - Interfaces ..... 647
  - IP ..... 117
  - IP address ..... 260
  - IP21 connector ..... 110
  - IP21OLEDB connector ..... 110
  - iSPCChart ..... 302
  - Iterator action block ..... 219
- ## J
- Java Database Connectivity (JDBC) ..... 27, 115
    - drivers* ..... 116, 118
  - Java Runtime Environment ..... 148
  - Java virtual machine (JVM) ..... 645
  - JavaRunFinalizer ..... 685
  - JavaRunGC ..... 686
  - JavaRunTimeStatus ..... 686
  - JavaScript ..... 30, 655
  - JavaThreadStatus ..... 686
  - JCOMessageList XML document ..... 282
  - JMS action block ..... 278
  - JMS message ..... 280
  - JNDI context provider ..... 278
  - JNDI lookup ..... 287
  - Join type ..... 244
  - Joined MDOs ..... 184, 191
    - attributes* ..... 192
    - join condition* ..... 192
  - Joiner action block ..... 243
  - JSON object ..... 254, 255
  - JSON to XML converter action block ..... 254
  - JSP page ..... 672
- ## K
- Key attribute ..... 188
  - Key figure ..... 289
  - Key performance indicator (KPI) .. 20, 481, 522
    - add* ..... 523
    - create* ..... 482
    - define* ..... 523
    - definitions* ..... 289
    - dimensions* ..... 485
    - lifecycle* ..... 486
    - relationships* ..... 486
    - scheduler* ..... 493
    - value* ..... 289
    - value maintenance* ..... 492
  - Keystore ..... 276
  - KPI monitor ..... 490, 491
  - KPI monitoring ..... 30
  - KPIConnector ..... 487
  - KPIQuery ..... 484
    - create* ..... 487
    - KPI connector* ..... 489

- KPIQuery (Cont.)
    - modes* ..... 489
  - KPIs and alerts action block ..... 296
- ## L
- Laboratory information management
    - system (LIMS) ..... 25–27
  - Lambda ..... 355
  - Language ..... 666
  - Language-specific properties ..... 397
  - LastErrorMessage ..... 263
  - LastName ..... 666
  - Layout configuration ..... 332
  - Legend ..... 342
  - Lighthammer Software ..... 16
  - Lights grid ..... 334, 337
  - Lightweight Directory Access Protocol (LDAP) ..... 43
  - Limit rows ..... 656
  - Line behavior ..... 579
  - Line chart ..... 345
  - Line monitor dashboard ..... 622, 638
  - Linear regression action block ..... 225
  - Link configuration ..... 205, 208, 226
  - Link editor ..... 206
  - Link types ..... 208
  - Load data ..... 307
  - Local file ..... 314
  - Local project ..... 146
  - Local properties ..... 203
  - Locale ..... 665
  - Localization ..... 396, 402
  - Log ..... 186
  - Log messages ..... 227
  - Log viewer ..... 229
  - Log/trace files ..... 227
  - Logging ..... 44, 291
  - Logging action block ..... 226
  - Logic action blocks ..... 209
  - Logical condition ..... 211
    - multiple inputs* ..... 212
  - Logical port ..... 544
  - Logical system ..... 543
  - Logon ticket ..... 266
  - Loops ..... 33
  - Lot size of one ..... 20
  - Lower alarms color ..... 362

## M

- Machine ..... 666
- Machine breakdown ..... 24
- Machine groups ..... 520
- Mail protocol ..... 312
- Mail server ..... 310
- MailAccount property ..... 310
- MailAccountPassword property ..... 310
- Maintenance notification ..... 25
- manifest.json file ..... 391
- Manufacturing dashboards action block .... 288
- Manufacturing data objects (MDOs) .... 32, 183
  - attributes* ..... 187
  - create* ..... 185
  - delete query* ..... 197
  - index* ..... 188
  - lifecycle* ..... 189
  - types* ..... 184
  - update query* ..... 197
  - visualization* ..... 190
- Manufacturing excellence ..... 24
- Manufacturing execution system (MES) ..... 25–27, 115
- Manufacturing process ..... 22
- Manufacturing synchronization ..... 24, 27
- Mapping ..... 206, 208
- Match mode ..... 335
- MatchValue property ..... 215
- Material availability ..... 28
- Material master ..... 511
- MaxIterations property ..... 218
- MDO service ..... 684
- MDOConnector data server ..... 184
- MDOQuery ..... 185, 194
  - join query* ..... 195
  - mode* ..... 194
  - query configuration* ..... 197
  - select query* ..... 195
- MeasureList ..... 169
- Measuring points ..... 445
- MEDIAN chart ..... 370
- MEDIAN-RANGE chart ..... 371
- Message action block ..... 235
- Message category ..... 140
- Message cleanup rules ..... 133, 141
- Message cleanup run interval ..... 142
- Message listeners ..... 28, 133, 135
- Message monitor ..... 28, 133, 139
- Message name ..... 140
- Message older than ..... 142

Message services ..... 27, 28, 131, 682  
     *actions* ..... 137  
 Message services action block ..... 281  
 Message status ..... 281  
 Message type ..... 140, 281, 547  
 MessageMimeType ..... 310  
 Messages ..... 102  
 MessageStatusText property ..... 284  
 Metadata XML ..... 266  
 Microsoft Internet Information Server ..... 16  
 Microsoft SQL Server ..... 115  
     *Analysis Services* ..... 116  
 Minitab export action block ..... 225  
 Minitab statistical software ..... 225  
 Miscellaneous functions action block ..... 222  
 Model-driven development ..... 27  
 Modeling area ..... 151  
 Modeling environment ..... 33  
 models.js ..... 394  
 Model-view-controller (MVC) ..... 383  
 Monitoring service ..... 683  
 MR chart ..... 375  
 MR-SHORT-RUN chart ..... 379  
 MS SQL Server ..... 27  
 Multidimensional Expressions (MDX)  
     query ..... 169  
 Multilights grid ..... 338  
 Multirowset XML ..... 244  
     *example* ..... 244

## N

Namespace queries ..... 157  
 Native mask ..... 177  
 Navigation  
     *configuration* ..... 409  
     *items* ..... 408  
 Navigation editor ..... 408  
 Near-real-time data access ..... 26  
 New directory ..... 315  
 Nodes ..... 439  
 Normalize action block ..... 244  
     *example* ..... 245  
 Notifications ..... 100  
     *example* ..... 105  
     *manage from SAP MII* ..... 105  
 NP chart ..... 373  
 Number format ..... 330  
 Number of inputs ..... 215

## O

OAG ..... 256  
 Object browser ..... 150  
 Object configuration ..... 205, 208  
 OData modeler ..... 198  
 OData service ..... 277  
 OData service action block ..... 276  
 OEEINT ..... 506, 570  
 OLAP ..... 27, 168  
 OLAP connector ..... 116  
 OLAPQuery template ..... 168  
     *configuration options* ..... 169  
 OLEDB connector ..... 121  
 On-demand MDOs ..... 184, 191  
 Open connector ..... 117  
 Operator dashboard ..... 602, 638  
     *alerts* ..... 605  
     *footer* ..... 605  
     *header* ..... 604  
     *key sections* ..... 604  
     *logo* ..... 605  
     *manage downtime* ..... 609  
     *manage orders* ..... 606  
     *menu* ..... 603  
     *plant maintenance notification* ..... 620  
     *report production* ..... 613  
     *shift review* ..... 617  
     *speed loss* ..... 615  
     *standard value key parameter* ..... 619  
 Oracle ..... 27  
 Oracle database ..... 115, 171  
 Order dispatch ..... 599  
 Order status ..... 594  
 Output XML property ..... 240  
 OutputMap property ..... 254  
 Overall equipment effectiveness  
     (OEE) ..... 35, 505

## P

P chart ..... 372  
 Parameter ..... 291  
     *input* ..... 202  
     *name* ..... 291  
     *output* ..... 202  
     *value* ..... 291  
 Parsing data ..... 214  
 Partner profile ..... 545  
 Passing variables ..... 668  
 Pause action block ..... 219  
 PCo connector ..... 127

PCoQuery ..... 122, 186  
 PCoQuery template ..... 159, 175  
     *modes* ..... 175  
     *TagAggregateQuery mode* ..... 179  
     *TagFixedQuery mode* ..... 180  
     *TagRetrieveQuery mode* ..... 178  
 PDF document ..... 318  
 Performance ..... 507  
 Permissions ..... 470  
 Persistency level ..... 186  
 Persistent MDOs ..... 184  
 Personal data manager ..... 66  
 PIC analysis query ..... 500  
     *dashboard creation* ..... 503  
     *nodes* ..... 502  
     *report* ..... 501  
 Pie chart ..... 346  
 Plant catalog hierarchy ..... 444  
 Plant data ..... 290  
 Plant floor connectivity ..... 106  
 Plant hierarchy ..... 535  
     *KPI targets* ..... 537  
     *machine groups* ..... 538  
     *resource details* ..... 536  
     *technical object* ..... 538  
 Plant hierarchy node ..... 512  
 Plant hierarchy template ..... 526  
     *node type* ..... 527  
     *relationship* ..... 527  
 Plant information catalog (PIC) ..... 413, 598  
     *call web services* ..... 464  
     *configure* ..... 499  
     *group* ..... 444, 448  
     *hierarchy* ..... 446  
     *map tag node* ..... 447  
     *plant data collection* ..... 468  
     *scheduler* ..... 445  
     *web services* ..... 454  
 Plant information catalog services ..... 682  
 Plant information maintenance ..... 426  
 Plant Information Maintenance  
     Service API ..... 456  
 Plant information maintenance services .. 681  
 Plant maintenance ..... 296, 442  
 Plant maintenance order ..... 25  
 Plant Manager 1.0 ..... 289  
 Plant Manager 2.0 ..... 289  
 Plant monitor dashboard ..... 622, 638  
 Plant network ..... 23  
 Plant-level reporting ..... 637  
 Pool connections ..... 261  
 Pooled connection ..... 79

POP ..... 312  
 Port ..... 117  
 Post message action block ..... 286  
 PostData property ..... 309  
 Postinstallation configurations ..... 53  
 Process data ..... 26  
 Process historian ..... 25  
 Process historian databases ..... 118  
 Process intelligence ..... 26  
 Process order ..... 628  
 Processing rules ..... 28, 133, 137  
 Processing status ..... 142  
 Production activities ..... 522  
 Production event template ..... 476  
     *create* ..... 477  
     *details* ..... 476  
     *open* ..... 476  
     *predefined templates* ..... 479  
     *restrictions* ..... 478  
     *weighting factor* ..... 478  
 Production events ..... 475  
     *create* ..... 479  
     *scheduler* ..... 481  
 Production modes ..... 521  
 Production order header list ..... 41  
 Production orders ..... 25, 28, 252, 513  
 Production planner ..... 25  
 Production planning ..... 442  
 Production supervisor ..... 24  
 ProfileEditor ..... 676  
 Programmable logic controller (PLC) ... 25, 175  
 Projects ..... 145, 151  
     *create* ..... 151  
 Properties ..... 150  
 Property resource bundles ..... 396  
 Property Set Service API ..... 463  
 Property sets ..... 414, 416, 418, 420  
     *create* ..... 425  
     *fields* ..... 420  
     *plant maintenance* ..... 421  
     *production planning* ..... 421  
 Property value ..... 207  
 Protocol ..... 271  
 Publish ..... 153

## Q

Quality ..... 507  
 Quality action block ..... 302  
 Query caching ..... 158, 160  
 Query messages action block ..... 281

- Query templates ..... 106, 155, 156, 490
    - connectors for a data buffer* ..... 161
    - create* ..... 156
    - data range configuration* ..... 162
    - data source configuration* ..... 157
    - general configuration* ..... 158
    - parameters configuration* ..... 166
    - properties* ..... 157
    - transformation configuration* ..... 166
    - types* ..... 168
  - Queue ..... 294
    - entries number* ..... 296
    - name* ..... 295
  - Queue delete action block ..... 296
  - Queue get action block ..... 295
  - Queue list action block ..... 296
  - Queue monitor ..... 571
  - Queue put action block ..... 295
  - Queue read action block ..... 280
  - Queue RFC (qRFC) ..... 264
  - Queue write action block ..... 278
  - Queuing action block ..... 294
  - Quick sizer tool ..... 40
- ## R
- Raise alert action block ..... 298
  - Read email action block ..... 312
  - Read message action block ..... 284
  - Read message as string action block ..... 286
  - Read timeout ..... 312
  - Real-time data synchronization ..... 26
  - Reason codes ..... 539
    - assignment to machine* ..... 597
    - configuration* ..... 581
    - levels* ..... 540
    - mass upload* ..... 540
  - Reference document
    - assign* ..... 257
  - Reference document loader action block ... 256
  - Reference documents ..... 256
  - Reference schema loader action block ..... 258
  - Refresh ..... 336
  - refresh() ..... 415
  - RegEx mask ..... 177
  - Releasing objects ..... 80
  - Remote folder ..... 314
  - Remote Function Call (RFC) .... 28, 132, 203, 261
    - destination* ..... 78, 542
    - execute* ..... 266, 268
    - names* ..... 262
  - Remote parent folder ..... 315
  - Remote procedure call (RPC) ..... 26
  - Remote terminal units (RTU) ..... 25
  - Render image ..... 305
  - Repeater ..... 653, 658
  - Repeater action block ..... 214
  - Repeater leg ..... 214
  - Repeating condition ..... 214
  - Repeating leg ..... 213
  - Report servlet ..... 398
  - Reports ..... 398, 669
  - Reprocessing ..... 139
  - Request XML property ..... 263
  - Rescheduling ..... 26
  - Response XML property ..... 263
  - REST service ..... 323
  - Result property ..... 223
  - Result set iterator action block ..... 220
  - ResultInNamespace ..... 117
  - ResultSet property ..... 220
  - Reverse consumption ..... 627
  - Reverse order ..... 629
  - RFC/BAPI ..... 33
  - RFC-enabled function modules ..... 657
  - RoleAttribList ..... 676
  - Role-based access ..... 673
  - RoleList ..... 677, 686
  - RoleProfile ..... 677
  - Roles ..... 154
  - Root sequence ..... 204
  - Root-cause analysis ..... 352
  - Row action block ..... 233
  - Rowset action block ..... 232
  - RSquared ..... 225
  - Rule description ..... 142
  - Rule-based ..... 137
    - processing* ..... 137
  - Run rule ..... 142
  - Runner service ..... 323
- ## S
- SALERT\_CREATE ..... 289
  - Sample size ..... 355
  - SAP alert action block ..... 289
  - SAP Business Technology Platform
    - (SAP BTP) ..... 198
  - SAP Business Warehouse (SAP BW) ..... 169
  - SAP BusinessObjects Business Intelligence
    - (SAP BusinessObjects BI) ..... 116, 169
  - SAP Digital Manufacturing Cloud for
    - insights ..... 640
  - SAP Enterprise Portal ..... 288

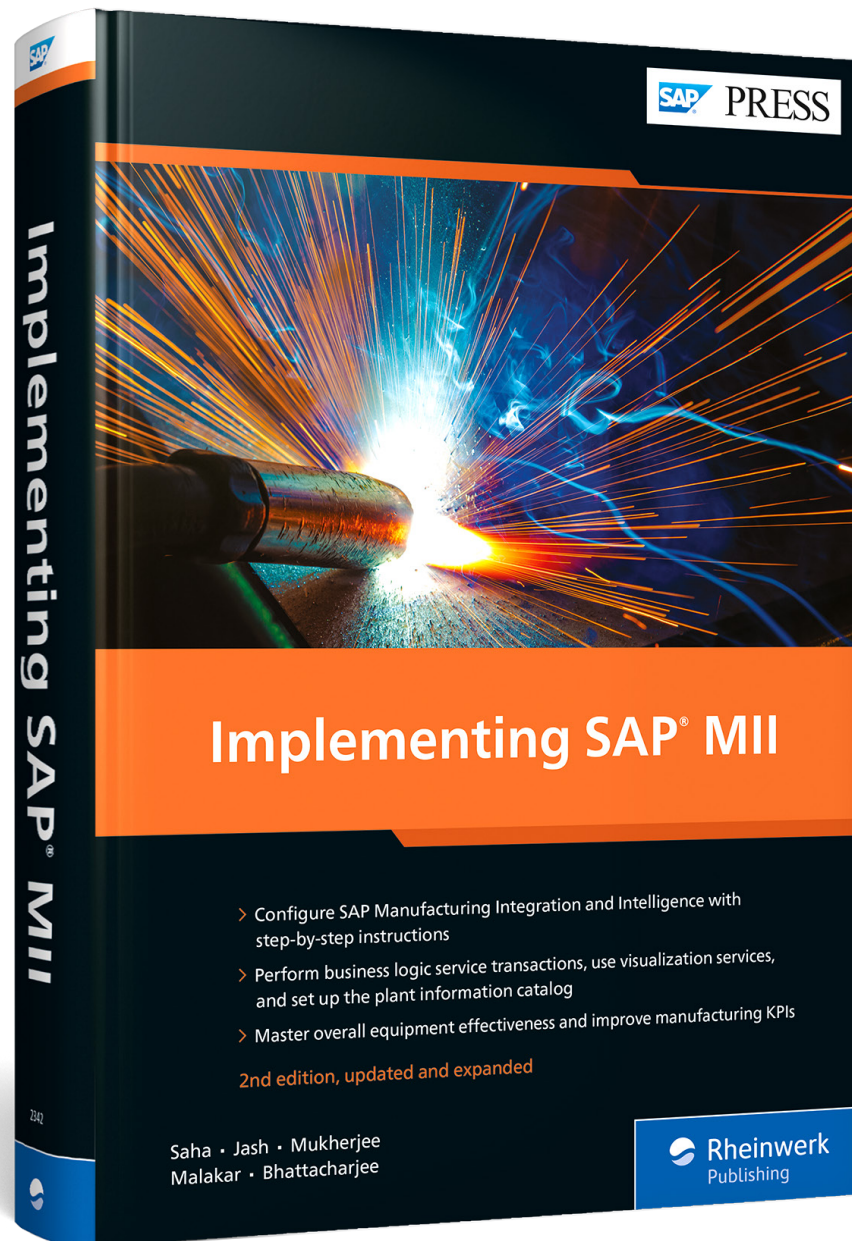
- SAP ERP ..... 27, 28, 252, 296
- SAP ERP system action block ..... 259
- SAP EWM ..... 553
- SAP HANA ..... 115, 198, 639
- SAP HANA smart data integration ..... 129
- SAP HANA Studio ..... 129
- SAP Java Connector ..... 27, 133, 260
  - configuration* ..... 260
  - session* ..... 263
- SAP Java Connector commit action
  - block ..... 265
- SAP Java Connector end session action
  - block ..... 265
- SAP Java Connector execute queue action
  - block ..... 264
- SAP Java Connector function action
  - block ..... 262, 266
- SAP Java Connector interface action
  - block ..... 261, 266, 659
- SAP Java Connector rollback action
  - block ..... 265
- SAP Java Connector start session action
  - block ..... 261
- SAP Java Resource Adapter ..... 45, 268
  - configuration* ..... 46
  - connection factory* ..... 269
- SAP Java Resource Adapter commit
  - action block ..... 270
- SAP Java Resource Adapter end session
  - action block ..... 270
- SAP Java Resource Adapter function
  - call action block ..... 269
- SAP Java Resource Adapter interface
  - action block ..... 268
- SAP Java Resource Adapter rollback
  - action block ..... 270
- SAP Java Resource Adapter start session
  - action block ..... 269
- SAP KPI update action block ..... 289
- SAP Lumira ..... 638
- SAP Manufacturing Execution (SAP ME) ... 476
- SAP MaxDB ..... 115
- SAP MII OLEDB UDS ..... 110
- SAP MII workbench ..... 28, 77, 145, 148, 491
  - content organization* ..... 151
  - layout* ..... 149
  - menu* ..... 152
  - objects* ..... 148
  - security* ..... 154
- SAP MII XML output action block ..... 229
- SAP NetWeaver Administrator ..... 229, 269
- SAP NetWeaver AS Java ..... 36, 39, 152, 281
  - administration* ..... 40
  - log* ..... 291
- SAP NetWeaver Developer Studio ..... 80
- SAP NetWeaver Development
  - Infrastructure (NWDI) ..... 36, 39, 66, 83, 146, 151
  - cockpit* ..... 72
  - configure* ..... 69
  - content development configuration* ..... 75
  - integration* ..... 73
  - wizard* ..... 70
- SAP OEE Archiving Configuration
  - template ..... 559
- SAP OEE Configuration template ..... 557
- SAP OEE Integration Configuration
  - template ..... 553
- SAP OEE Management ..... 34, 40, 190, 476, 505
  - authorizations* ..... 509
  - component architecture* ..... 506
  - data model* ..... 511
  - overview* ..... 506
  - post-installation* ..... 513
  - user roles* ..... 510
- SAP OEE Management dashboard ..... 602
- SAP OEE Migration template ..... 558
- SAP OEE Model Plant Data Setup ..... 559
- SAP OEE NetWeaver Configuration
  - template ..... 551
- SAP Plant Connectivity ..... 27, 36, 41, 121, 176, 177
  - add destination system* ..... 89
  - agent instance* ..... 91
  - components* ..... 48
  - create source system* ..... 86
  - destination system* ..... 88
  - destination system configuration* ..... 90
  - installation* ..... 47
  - management console* ..... 52
  - postinstallation* ..... 50
  - security* ..... 90
  - server sizing* ..... 47
  - source system* ..... 84
  - use* ..... 83
  - Windows configuration* ..... 49
- SAP Plant Connector ..... 162
- SAP PO integration directory ..... 271
- SAP S/4HANA ..... 27, 28, 136, 252, 296, 475
  - connect* ..... 440
  - customization* ..... 514
  - display master data* ..... 600
  - master data setup* ..... 529

SAP S/4HANA (Cont.)  
*source assignment* ..... 443  
*standard master data* ..... 530  
SAP xMII ..... 16  
SAP xMII 12.0 ..... 16  
SAP\_Commons ..... 420  
SAP\_OEE\_PH\_Node ..... 425  
SAP\_ShiftPattern property ..... 500  
SAP\_Tariff property ..... 499  
SAPServerInfo ..... 686  
SAPUI5 ..... 36, 382  
*application development* ..... 382  
*controller* ..... 386  
*files* ..... 384  
*folder structure* ..... 389  
*HTML* ..... 388  
*library* ..... 389  
*localization* ..... 396  
*mobile devices* ..... 400  
*project creation* ..... 383  
*view* ..... 385  
*viz frame pie chart* ..... 385, 386  
SAPUserName property ..... 267  
SAPVariableList ..... 169  
Scaled height ..... 305  
Scaled width ..... 305  
Scaling factor ..... 247  
Scatter chart ..... 349  
ScheduleAttribList ..... 686  
Scheduled downtimes ..... 598  
Scheduled jobs ..... 435  
ScheduleDetailList ..... 686  
ScheduleList ..... 686  
Scheduler ..... 189, 324  
*history* ..... 326  
Scheduler services ..... 678  
Schedules ..... 163  
Schema elements ..... 258  
Scoreboard grid ..... 337  
Scrape HTML page action block ..... 309  
*example* ..... 309  
Search pattern ..... 309  
Search template ..... 491  
*create* ..... 491  
*delete* ..... 492  
*run* ..... 492  
Security ..... 40, 153  
Self-service composition  
environment (SSCE) ..... 403, 475, 503  
*Design Dashboard tab* ..... 404  
*icons* ..... 406  
*launch dashboard* ..... 407  
Self-service composition environment (SSCE)  
(Cont.)  
*layout objects* ..... 406  
*My Dashboard tab* ..... 403  
*Source Code Editor tab* ..... 407  
Send mail action block ..... 310  
Sender HTTP adapter ..... 271  
Sender SOAP adapter ..... 273  
SendTimeout property ..... 310  
Sequences ..... 203  
*add* ..... 204  
*insert* ..... 204  
Server aliases ..... 260, 261  
Server Name ..... 140  
ServerAttribList ..... 687  
ServerInfo ..... 687  
ServerList ..... 686  
ServerURL ..... 114, 115, 118  
ServiceList ..... 687  
Session variables ..... 127, 664, 668  
SessionList ..... 677  
Set alert status action block ..... 300  
setColumns ..... 656  
setDisplayTemplate(TEMPLATENAME) ..... 415  
SFTP action block ..... 320  
SFTP create directory action block ..... 321  
SFTP delete file action block ..... 321  
SFTP get action block ..... 320  
SFTP get file list action block ..... 321  
SFTP put action block ..... 321  
Shared project ..... 146  
Shift patterns ..... 497, 498  
*create* ..... 498  
Shifts ..... 530  
*definitions* ..... 530  
*sequences* ..... 531  
*work center* ..... 532  
Shop floor automation ..... 21  
Shop floor execution ..... 22  
Show title ..... 354  
Simple statistics action block ..... 224  
Simulator connector ..... 122  
Simulator data server ..... 122  
Simulator editor ..... 122  
Simulator groups ..... 122  
Simulator instance ..... 122  
Simulator tags ..... 122  
Single rowset  
XML ..... 244  
Single sign-on (SSO) ..... 266  
Six Sigma analysis ..... 30

Sizing determination ..... 40  
*parameters* ..... 41  
Slope ..... 225  
Sorting data ..... 33, 275, 323  
Software components ..... 39, 68, 74  
Sorting order ..... 657  
Sorting order ..... 242  
Source control ..... 77  
Source document ..... 246  
Source for upper control limit ..... 361  
Source of parent document ..... 243  
Source system ..... 84  
SPC/SQC analysis ..... 33  
SPC/SQC analysis action nblock ..... 302  
SPC/SQC chart action block ..... 302  
Specification limits ..... 359  
SQC analysis XML ..... 303  
SQC indexes ..... 302  
SQL ..... 27  
SQL queries ..... 115  
SQL transaction begin action block ..... 293  
SQL transaction clear batch action block ..... 294  
SQL transaction commit action block ..... 293  
SQL transaction end action block ..... 294  
SQL transaction execute batch action  
block ..... 294  
SQL transaction rollback action block ..... 294  
SQL transaction support action block ..... 293  
SQLQuery template ..... 157, 170, 194  
*configuration screens* ..... 171  
*example* ..... 172  
*expressions* ..... 171  
*modes* ..... 170  
SQLTransaction property ..... 293  
SSCC barcode ..... 626  
Stacked bar chart ..... 347  
Standard deviation ..... 180  
Standard rate ..... 537  
Statistical analysis ..... 33  
Statistical indexes ..... 224, 236  
Statistical methods ..... 176  
Statistical process control (SPC) ..... 30  
*chart* ..... 351  
*chart servlet* ..... 399  
*rules* ..... 362  
Statistical quality control (SQC) ..... 30, 351  
Statistics service ..... 679  
Status ..... 687  
String list ..... 249  
String list to XML parser action block ..... 249  
String to XML parser action block ..... 248  
StringTypeIndicator ..... 119  
Strip quotes ..... 249  
Subgroup size ..... 355  
Subject ..... 310  
Supervisory control and data  
acquisition (SCADA) ..... 24, 26, 27, 288  
SVG documents ..... 304  
SVG renderer action block ..... 304  
Switch action block ..... 215  
Synchronize ..... 131  
Synchronous communication ..... 271  
System Landscape Directory (SLD) ..... 67  
System properties ..... 53  
*cross-origin resource sharing* ..... 61  
*general settings* ..... 53  
*limits* ..... 54  
*proxy configuration* ..... 60  
*transaction engine* ..... 55  
*tuning* ..... 57  
System setup ..... 39  
SystemInfo service ..... 685  
SystemList ..... 677  
**T**  
Tab-delimited file ..... 225  
Tag groups ..... 118, 176  
Tag node ..... 431, 438  
Tag queries ..... 110, 114, 122  
Tag store query ..... 178  
Tag-based relational databases ..... 117  
TagInfoDescriptionColumn ..... 111, 119  
TagInfoMaxRangeColumn ..... 111, 120  
TagInfoMinRangeColumn ..... 111, 120  
TagInfoNameColumn ..... 111, 119  
TagInfoQuery ..... 120  
TagInfoTable ..... 111  
TagInfoTypeColumn ..... 111, 120  
TagList ..... 118  
TagList mode ..... 119  
TagListDescriptionColumn ..... 111, 120  
TagListNameColumn ..... 111, 120  
TagListQuery ..... 118, 120  
TagListTable ..... 110  
Tags ..... 176, 431  
*catalog query* ..... 435  
*general attributes* ..... 433  
*log* ..... 438  
*map with data server* ..... 436  
*property sets* ..... 436  
*transaction* ..... 434  
Tariff group ..... 496  
Template categories ..... 150

- Terminate transaction action block ..... 216
  - Termination message ..... 216
  - Text loader action block ..... 307
  - Text saver action block ..... 307
  - Throw action block ..... 222
  - Time ..... 247
  - Time interpolator action block ..... 248
  - Time period schedules ..... 497
  - Time periods ..... 163, 497
    - configuration* ..... 163
    - schedules* ..... 164, 165
  - Time series data ..... 176
  - Time weighted average ..... 180
  - TimePeriodAttribList ..... 687
  - TimePeriodList ..... 687
  - ToAddress property ..... 310
  - Token-based authentication ..... 276
  - Tolerance type ..... 356
  - Topic publish action block ..... 279
  - Totalizer action block ..... 246
  - Totalizer function ..... 180
  - Totalizer mode ..... 247
  - Trace and log ..... 226
  - Trace locations ..... 45
  - Tracer action block ..... 226
  - Tracing ..... 44
  - Transaction
    - /BD54* ..... 543
    - /OEE/GLOBAL\_HIER* ..... 534
    - /OEE/PLANT\_RC* ..... 539
    - /OEE/RC\_EXCELUPLOAD* ..... 540
    - /WE20* ..... 545
    - /WE21* ..... 544
    - ALRTCATDEF* ..... 289
    - BD64* ..... 549
    - CRO2* ..... 532
    - SM59* ..... 135
  - Transaction call action block ..... 217
  - Transaction manager ..... 327
  - Transaction properties ..... 127
  - Transaction service ..... 680
  - Transactions ..... 33, 201
    - create* ..... 634
    - execute* ..... 208
    - execute without trace* ..... 208
    - handling* ..... 261
    - properties* ..... 202
    - thread* ..... 327
  - Transform ..... 249
  - Transitions ..... 247
  - Transport service ..... 680
  - Trigger expression ..... 101
  - Trim whitespace ..... 249
  - TripleDES ..... 65
- ## U
- U chart ..... 374
  - UDDI registry ..... 276
  - Union action block ..... 246
  - UniqueName ..... 665
  - Unit conversion ..... 224
  - Unit converter action block ..... 223
  - Unit of measure (UOM) ..... 224
  - Universal connectivity ..... 27, 29
  - Update message action block ..... 284
  - Upper alarms color ..... 362
  - Upper chart area configuration ..... 359
  - Upper control limit ..... 361
  - UpTime ..... 687
  - UseOLEDB ..... 114
  - User group assignment ..... 573
  - User management ..... 42
    - create new* ..... 43
  - User Management Engine (UME) ..... 42, 146, 408, 551
  - User roles ..... 42
  - UserAttribList ..... 677
  - UserList ..... 677, 687
  - UserProfile ..... 677
  - UTF-8 ..... 313
- ## V
- Validate XML action block ..... 250
  - Value lookup action block ..... 254
  - Value map ..... 251, 254
  - Value map action block ..... 252
  - ValueSource property ..... 237
  - Virtual server connectors ..... 124, 662
  - Virtual servers ..... 123, 662
  - Virtual tables ..... 130
  - Virtual tag ..... 433
  - VirtualIDBC connector ..... 662
  - Visibility and responsiveness ..... 20
  - Visualization engine ..... 27
  - Visualization services ..... 29, 329
  - Visualizations and analytics ..... 30
- ## W
- Web action block ..... 306
  - Web content
    - localizing* ..... 401

- Web report development ..... 382
  - Web service ..... 201
  - Web service action block ..... 274
  - Web service XI action block ..... 273
  - Web services ..... 26
  - Web Services Description
    - Language (WSDL) ..... 273
    - URL* ..... 274, 276
  - Web view ..... 151
  - web:// ..... 307
  - WebService ..... 117
  - Western Electric Company (WECO) ..... 362
    - rule numbers* ..... 359, 362
    - rules* ..... 362
  - What-if analysis ..... 359
  - While loop action block ..... 218
  - Windows service ..... 92
  - Wonderware IndustrialSQL Server ..... 114
  - Work break schedules ..... 530
  - Worker UI ..... 561
    - behavior* ..... 574
    - general configuration* ..... 572
    - global configuration* ..... 561
    - supported plants* ..... 562
  - Workflows ..... 563
    - correlation* ..... 567
    - split* ..... 566
    - standard* ..... 564
  - Workfile action block ..... 313
  - WSDLGen ..... 324
- ## X
- Xacute connector ..... 127
  - Xacute data server ..... 127
  - XacuteConnector ..... 127, 665
  - XacuteQuery ..... 127, 217, 290, 321, 334, 388, 667
  - XBAR chart ..... 364
  - XBAR-MR chart ..... 365
  - XBAR-RANGE chart ..... 365, 379
  - XBAR-SDEV chart ..... 366
  - XDP file ..... 319
  - XI action block ..... 271
  - XMIIMESSAGELISTENER ..... 134
  - XML ..... 122, 202, 229
    - attributes* ..... 208
    - data type* ..... 322
    - HTTP service* ..... 201
    - join conditions* ..... 244
  - XML connector ..... 126
  - XML document ..... 229, 231, 232, 255, 257, 316
  - XML functions action block ..... 236
  - XML loader action block ..... 307
  - XML parsing ..... 33
  - XML saver action block ..... 307
  - XML Schema Definition (XSD) ..... 250
    - document* ..... 258
  - XML style sheet (XSL) ..... 167, 249
  - XML to JSON converter action block ..... 255
  - XML tracer action block ..... 227
    - output example* ..... 228
  - XML transformation ..... 41
  - XMLA ..... 116
  - XMLA data source ..... 116
  - XMLQuery template ..... 181
  - XPath ..... 167, 651, 653
  - XPath expression ..... 214, 236, 241, 252
  - XSL transformation ..... 167
  - XSL transformation action block ..... 249
  - XSLT ..... 566
  - XSLT library ..... 167
- ## Z
- ZIP file name ..... 315



Saha, Jash, Mukherjee, Malakar, Bhattacharjee

## Implementing SAP MII

705 pages | 01/2023 | \$99.95 | ISBN 978-1-4932-2342-8

 [www.sap-press.com/5610](http://www.sap-press.com/5610)



**Dipankar Saha** is an architect for SAP manufacturing and Industry 4.0 solutions and has worked in several SAP MII implementation projects and also in the product design and development of SAP MII. Dipankar is an SAP-certified associate enterprise architect. He has more than 20 years of experience in the IT industry and has a bachelor's degree in chemical engineering.



**Chandan Jash** is a technical architect and SME for SAP ME and SAP MII implementation projects at IBM India. He has worked on many SAP manufacturing implementations, supported SAP MII and SAP ME projects globally in different roles, and has implemented Industry 4.0 standards. He has worked with SAP solutions for more than 17 years and has a bachelor's degree in computer science and engineering.



**Sudipta Mukherjee** is an application architect at IBM India with more than 17 years of IT experience. His main areas of expertise are SAP MII, SAP OEE Management, SAP ME, SAP Digital Manufacturing Cloud, SAP BTP, and SAP Internet of Things. He works with global clients to define SAP manufacturing implementation architectures and manage end-to-end delivery. He has a bachelor's degree in computer science and engineering.



**Supriya Malakar** is a senior consultant at IBM India. He works with global clients for large-scale SAP MII and SAP ME solution implementation. He also has deep expertise in SAP Business Technology Platform and SAP Digital Manufacturing Cloud. He has more than six years of experience in the IT industry and has a bachelor's degree in computer science and engineering.



**Abesh Bhattacharjee** has been working as an IT consultant, a go-to market leader, and an architect for more than 20 years. He is currently the senior solution architect at Neptune Software responsible for North America.

*We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.*