Perceptive Connector for SAP Financials

Installation and Setup Guide

Version: 2.8.x

Written by: Documentation Team, R&D
Date: April 2022

Copyright

Table of Contents

What is Perceptive Connector for SAP Financials?

Perceptive Connector for SAP Financials creates a data channel between Perceptive products and your SAP application. The connector contains operations that allow you to create, park, post invoices and credit memos in SAP, and perform live lookups from SAP for data entry and validation.

Perceptive Connector for SAP Financials communicates with the SAP Business Application Programming Interface (BAPI). These BAPI calls allow data exchange with SAP using SAP Java Connector (SAP JCo).

The connector relies upon Perceptive Connect Runtime, which provides independent messaging and data exchange between applications. It allows you to configure, create, and maintain operation mapping without the need to access the file system.

Overview of the set up process

To install and configure Perceptive Connector for SAP Financials, complete the procedures.

Prerequisite Perceptive Connect Runtime is installed on your system.

1. Install the connector
2. Configure Perceptive Content to use the connector
3. Create and configure a channel

About installing the connector

The PerceptiveConnectorForSAPFinancials-<version>.zip file includes the files required to integrate Perceptive Connector for SAP Financials, Perceptive Content, and your SAP application.

You can install Perceptive Connector for SAP Financials and related files manually using Perceptive Connect Runtime Web Console or from the “Upload new bundles” section in Perceptive Connect Runtime. You must install the connector on the same computer as the Perceptive Connect Runtime service.

Download the required files from the Customer Portal

To download the ZIP file, complete the following steps.

1. Log into the Customer Portal.
2. In the **Product Downloads** page, search for the **PerceptiveConnectorForSAPFinancials-<version>.zip** file.
3. Download the file in the temporary directory. Your computer must have access to **Perceptive Connect Runtime Web Console**, located at **http://<*Perceptive Connect Runtime machine name*>:<*port*>**.

Install the connector

Install the connector using the drag-and-drop feature in Perceptive Connect Runtime

To install Perceptive Connector for SAP Financials in Perceptive Connect Runtime, complete the following steps.

1. In a browser, type the Web Console address in the format `http://<Perceptive Connect Runtime machine name>:<port>`.
2. Click **Install a Connector**.
3. Locate the **ConnectorBundles_32_bit-<version>.zip** or **ConnectorBundles_64_bit-<version>.zip** file from the temporary directory where **PerceptiveConnectorForSAPFinancials-<version>.zip** file is extracted.
4. In **Perceptive Connect Runtime**, on the **Upload new bundles** page, drag the **ConnectorBundles_32_bit-<version>.zip** file if you have Java Runtime Environment (32-bit) installed, or **ConnectorBundles_64_bit-<version>.zip** file if you have Java Runtime Environment (64-bit) installed, to the **DRAG FILES HERE** box.

   Result **Perceptive Connect Runtime** extracts and installs the bundles included in the ZIP file automatically and displays the number of installed bundles in the **Pending** box.
5. In the **Pending** box, click **Accept** to complete the installation.

   Result The installation summary appears in the **Completed** box.
6. Optional. If you integrate Perceptive Connector for SAP Financials with Perceptive Content 7.2.2, you must complete the following steps.
   1. In **[*Drive*:]\<*pcr_installation_directory*>\conf**, open the **config.properties** file.
   2. Add **sun.*** at the end of the value of the **org.osgi.framework.bootdelegation** property.The value must be `javax.xml.*,org.xml.sax.*,org.w3c.dom,org.xml.sax,com.sun.*,sun.*`
   3. In **[*Drive*:]\inserver\etc**, open the **inow.ini** file.
   4. Under **[Envoy]**, update the property for **pcr.soap.bridge.url** to `http://<ip_of_pcr>:<port_of_pcr>/rs/soapBridge?_wadl`
   5. Restart **ImageNow Server 7.2** service.
   6. Open the **inserverWorkflow.ini** file and update the **connect.uri** property to `http://<ip_of_pcr>:<port_of_pcr>/rs/workflowTrigger?_wadl`
   7. Restart the **ImageNow Workflow Agent 7.2** service.

Install the connector manually using Perceptive Connect Runtime

To install Perceptive Connector for SAP Financials manually, complete the following steps.

Prerequisite You must extract the PerceptiveConnectorForSAPFinancials-<version>.zip file to a temporary directory.

1. In a browser, type the Web Console address in the format `http://<Perceptive Connect Runtime machine name>:<port>/connect`.

Note The Perceptive Connect Runtime service runs on port 80, by default. Each instance of Perceptive Connect Runtime that runs on a single host machine requires a unique port number.

2. In the browser dialog box, enter your user name and password. The default user name is admin and the default password is admin.
3. In Perceptive Connect Runtime Web Console, under **Perceptive Connect**, click **View Bundles**.
4. In the **Bundles** page click **Install/Update** and complete the following steps.
    1. In the **Upload/Install Bundles** dialog box, click **Choose File**, navigate to the **ErpCommon-<version>.jar** file, and click **Open**.
    2. Select the **Start Bundle** check box, set the **Start Level** to **4** and click **Install or Update**.
    3. Restart Perceptive Connect Runtime service.

       Note Install the **ErpCommon-<version>.jar** file before you install the other bundles because they are dependent on the **ErpCommon-<version>.jar** file.

       Based on the Java version (32-bit/64-bit) running on your Windows system, install either the **SapJco3_win32-<version>.jar** or the **SapJco3_win64-<version>.jar** file after installing the **ErpCommon-<version>.jar** file.
5. To install the JAR files included in the connector ZIP file, repeat the following substeps for each file.
    1. Click **Browse** and navigate to the connector in your temporary directory.
    2. Click a connector JAR file and click **Open**.
    3. Click **Install or Update**.

       Important Install **ErpImplSap-<version>.jar** file after installing the **ErpCommon-<version>.jar** and 32-bit or 64-bit version of the **SapJco3_<*32-bit or 64-bit version of Windows operating system*>_<version>.jar** file.
6. Optional. If you integrate Perceptive Connector for SAP Financials with Perceptive Content 7.2.2, you must complete the following steps.
    1. In **[*Drive*:]\<*pcr_installation_directory*>\conf**, open the **config.properties** file.
    2. Add **sun.*** at the end of the value of the **org.osgi.framework.bootdelegation** property.The value must be `javax.xml.*,org.xml.sax.*,org.w3c.dom,org.xml.sax,com.sun.*,sun.*`
    3. In **[*Drive*:]\inserver\etc**, open the **inow.ini** file.
    4. Under **[Envoy]**, update the property for **pcr.soap.bridge.url** to `http://<ip_of_pcr>:<port_of_pcr>/rs/soapBridge?_wadl`
    5. Restart **ImageNow Server 7.2** service.
    6. Open the **inserverWorkflow.ini** file and update the **connect.uri** property to `http://<ip_of_pcr>:<port_of_pcr>/rs/workflowTrigger?_wadl`
    7. Restart the **ImageNow Workflow Agent 7.2** service.

Reconfigure the connector with upgraded Perceptive Content 7.2.2

If your Perceptive Connector for SAP Financials <version> is configured with Perceptive Content that is now upgraded to version 7.2.2, you must reconfigure the connector to work with this upgrade.

Reconfigure the connector

To reconfigure Perceptive Connector for SAP Financials <version> to work with Perceptive Content 7.2.2, you must complete the following steps.

Prerequisite Install Perceptive SOAPBridge Connector, version 1.0.5

1. In **[*Drive*:]\<*pcr_installation_directory*>\conf**, open the **config.properties** file.
2. Add **sun.*** at the end of the value of the **org.osgi.framework.bootdelegation** property.The value must be `javax.xml.*,org.xml.sax.*,org.w3c.dom,org.xml.sax,com.sun.*,sun.*`
3. In **[*Drive*:]\inserver\etc**, open the **inow.ini** file.
4. Under **[Envoy]**, update the property for **pcr.soap.bridge.url** to `http://<ip_of_pcr>:<port_of_pcr>/rs/soapBridge?_wadl`
5. Restart **ImageNow Server 7.2** service.
6. Open the **inserverWorkflow.ini** file and update the **connect.uri** property to `http://<ip_of_pcr>:<port_of_pcr>/rs/workflowTrigger?_wadl`
7. Restart the **ImageNow Workflow Agent 7.2** service.

Upgrade the connector

You can upgrade the connector from a previous version using the drag-and-drop feature in Perceptive Connect Runtime or you can extract and install the JAR files manually. If you upgrade from a version lower than Perceptive Connector for SAP Financials 2.5.x, you must add the function `customAccountInvoiceDataMassager(invoice, logger) {}` to your custom scripts that are located in the [*drive*:]/{*Perceptive Connect Runtime installation directory*}/SAPCustomScript directory. You can also use the web interface to add the function to your custom script by navigating to the address `<IP address of the Perceptive Connect Runtime>:<port number>/sfc/dataMassaging.html` in your web browser.

After you upgrade, create child parameters under InvoiceLineItemList for PO1 to PO10 fields in the mapping XML of the CreateInvoice and ParkInvoice operations and map them manually to blank values or eForm fields. These mappings enable the connector to perform customized conditional checks using a data massaging script on fields retrieved from the GetPurchaseOrderDetails operation when you perform a CreateInvoice or ParkInvoice operation.

Upgrade using the drag-and-drop feature in Perceptive Connect Runtime

To upgrade the connector using the drag-and-drop feature in Perceptive Connect Runtime, complete the following steps.

1. In a browser, type the Web Console address in the format `http://<Perceptive Connect Runtime machine name>:<port>`.
2. Click **Install a Connector**.
3. Locate the **ConnectorBundles_32_bit-<version>.zip** or **ConnectorBundles_64_bit-<version>.zip** file from the temporary directory where **PerceptiveConnectorForSAPFinancials-<version>.zip** file is extracted.
4. In **Perceptive Connect Runtime**, on the **Upload new bundles** page, drag the **ConnectorBundles_32_bit-<version>.zip** file if you have Java Runtime Environment (32-bit) installed,

or **ConnectorBundles_64_bit-<version>.zip** file if you have Java Runtime Environment (64-bit) installed, to the **DRAG FILES HERE** box.

Step result: **Perceptive Connect Runtime** extracts and installs the bundles included in the ZIP file automatically and displays the number of installed bundles in the **Pending** box.

5. In the **Pending** box, click **Accept** to complete the installation.

   Result The installation summary appears in the **Completed** box.

Upgrade manually using Perceptive Connect Runtime

To upgrade the connector by manually uninstalling and installing the connector, complete the following steps.

1. In **Perceptive Connect Runtime Web Console**, under **Perceptive Connect**, click **View Bundles**, and in the **Bundles** page, locate the following bundles.
   - ErpCommon
   - SapJco3_win32 or SapJco3_win64 (based on Java Runtime Environment 32-bit/64-bit installed on your computer)
   - ErpImplSap
   - ErpAsyncConnector
   - ErpSyncConnector
   - ErpImplCache
   - ErpDiag
2. Click the **Uninstall** 🗑 button next to each bundle to uninstall it.

   Note If you upgrade from a version earlier than Perceptive Connector for SAP Financials 2.0.0, you must also uninstall Perceptive Connect and install Perceptive Connect Runtime.

   Important Uninstalling Perceptive Connect also uninstalls all Perceptive products that are installed using Perceptive Connect. If there are other Perceptive products installed that use Perceptive Connect, you need to reinstall their Perceptive Connect Runtime compatible versions and configure them. For additional information on how to install and configure other Perceptive products, refer to their respective installation and configuration guides.
3. Reinstall each of the bundles using the Perceptive Connect Runtime.
4. Under **Perceptive Connect,** click **View Bundles**, and in the **Bundles** page, click the **Refresh Package Imports** ↻ button next to each reinstalled bundle.
5. Click the **Update** ⇄ button next to each reinstalled bundle.
6. Click the **Perceptive Connect** menu, and then click **View Configuration**.
7. Under **ERP SAP Config,** open each of the SAP configurations and click **Save**.

   Note After installing Perceptive Connector for SAP Financials, if you required to update the ErpAsyncConnector-<version>.jar or ErpSyncConnector-<version>.jar file in the Perceptive Connect Runtime, ensure that you unbind and save the respective configuration settings.

Set up SAP connectivity

About setting up SAP connectivity

After you install Perceptive Connector for SAP Financials, you need to configure Integration Server, create one or more SAP connections, and perform additional configuration based on your business requirement.

Perceptive Connector for SAP Financials includes the following bundles and operations.

Bundle name	Description	Operation
ErpCommon-<version>.jar	All other bundles of Perceptive Connector for SAP Financials are dependent on ErpCommon. Hence, install this bundle before you install the rest of the bundles.	
SapJco3_win32-<version>.jar	SapJco3_win32 bundle is required to install the Connector for SAP Financials in computers running on Windows operating system with Java 32-bit installed.	
SapJco3_win64-<version>.jar	SapJco3_win64 bundle is required to install the Connector for SAP Financials in computers running on Windows operating system with Java 64-bit installed.	
ErpAsyncConnector-<version>.jar	In ErpAsyncConnector, all the operations except CheckStatus are asynchronous calls. Only CheckStatus is a synchronous call in this bundle.	ParkInvoice CreateInvoice PostInvoice CancelInvoice CheckStatus
ErpImplSap-<version>.jar	ErpImplSap bundle is required to communicate with SAP.	

Bundle name	Description	Operation
ErpSyncConnector-<version>.jar	All operations of ErpSyncConnector are synchronous calls.	GetCompanyDetails GetCompanyList GetCostCenterDetails GetCostCenters GetCurrenciesForCountry GetCurrencyList GetEmailAddresses GetErpServiceList GetGLAccounts GetPaymentTermList GetProfitCenterList GetPurchaseOrderDetails GetPurchaseOrderList GetPurchaseRequisitionDetails GetTaxCodes GetVendorDetails GetVendorList GetVendorRemitToList
ErpImplCache-<version>.jar	ErpImplCache bundle is required for caching data obtained from SAP.	
ErpDiag-<version>.jar	ErpDiag bundle is required to run the diagnostic utility.	
ErpDataMassagingUIComponent-<version>.jar	ErpDataMassagingUIComponent is required to include the web interface that enables you to configure data massaging.	

Configure Integration Server using the web console

To provide Integration Server credentials in Perceptive Connect Runtime Web Console, complete the following steps.

1. Log in to **Perceptive Connect Runtime Web Console**.
2. Under **Perceptive Connect**, click **View Configuration** and in the **View Configuration** page, in the **Perceptive Content Connector** section, click the **Edit the configuration values** ✎ button next to **Connection Manager**.

3. In the **Connection Manager** dialog box, complete the following steps.
    1. In the **Connection Provider Target** list, select **Integration Server 6.7**.
    2. In the **User Name** box, type the user name you use to log into Perceptive Content.
    3. In the **Password** box, type your password.
    4. Click **Save**.

Configure SAP connections

You can configure multiple SAP connections in the connector. To configure the SAP connection, complete the following steps.

1. In **Perceptive Connect Runtime Web Console**, under **Perceptive Connect**, click **View Configuration**.
2. In the **View Configuration** page, under **General**, click the **Create new factory configuration** + button next to **ERP SAP Config**.
3. In the **ERP SAP Config** dialog box, in **Unique ID** box, type a unique name for the SAP connection and in the **Connection Type** list, select the type of server for connection.
    - For a **Custom Application Server** connection, type the SAP application server host name or IP address in the **AS Host** box, and the number of instances to connect to SAP servers in the **Instance Number** box. **AS Host** and **Instance Number** are required properties.

      Note If the SAP gateway is located on a system that uses NAT IP address, type the virtual host name of SAP application server.
    - For a **Group/Server Selections** connection, type the SAP message server host name or IP address in the **MS Host** box, the message server group in the **Group** box, and the message server R3 name in the **R3 Name** box. **MS Host**, **Group**, and **R3 Name** are required parameters.

      Note If the SAP gateway is located on a system that uses NAT IP address, type the virtual host name of SAP message server.
4. In the **ERP SAP Config** dialog box, complete the following additional steps.
    1. Optional. In the **SAP Router** box, provide the SAP router string if you want to access the SAP application using the SAP router.
    2. In the **User Name** box, type the user name and in the **Password** box, type the corresponding password to log into your SAP application.
    3. In the **Client** box, type the client for the SAP application.
    4. In the **Language** box, specify the language. For example, type `en` for English.
    5. In the **Pool Capacity** box, enter the maximum number of idle connections you want to keep open. The default value is 2. Contact your administrator if you cannot complete this step.
    6. In the **Peak Limit** box, enter the maximum number of connections that can be active simultaneously. The default value is 10 and minimum is 2. Contact your administrator if you cannot complete this step.

       Note If you set a very high peak limit, the connection may take longer to connect with SAP server.
    7. In the **Expiration Time** box, specify the time in seconds after which the idle connections close. The default value is 300 seconds. Contact your administrator if you cannot complete this step.

8. In the **Expiration Time Check Period** box, specify the check interval in seconds for expired connections. The default value is 30 seconds. Contact your administrator if you cannot complete this step.
9. Optional. In the **Data Delimiter** box, specify a character as delimiter to replace the default delimiter "|". If you enter multiple characters in this field, the first character is considered as the delimiter.
10. Optional. In the **Connector Date Format** box, select a date format that you want to enforce on synchronous or asynchronous operations when they input a date to the Perceptive Connector for SAP Financials. By default a date format is not enforced.
11. Optional. In the **Duplicate Invoice Check Config ID** box, provide the **DupCheck ID**, and in the **ASQ Behavior for Duplicate Invoice** box, select either **Route To Success Queue (Soft Stop)** or **Route To Failure Queue (Hard Stop)** option to configure duplicate invoice check.

    Note Before you provide the **DupCheck ID**, you must configure SAP duplicate invoice check and create the **DupCheck ID**. If you select the **Route To Success Queue (Soft Stop)** option, duplicate invoice check is populated as TRUE but Invoice number and fiscal year is not populated. If you select the **Route To Failure Queue (Hard Stop)** option, no details are populated.
12. Optional. In the **Item categories for accounting data population** box, specify item categories for which accounting data must be passed to SAP while performing ParkInvoice or CreateInvoice operations.

    Note If you specify multiple item categories, use semicolon to separate them.
13. Optional. In the **WBS element conversion required** check box, specify whether WBS element conversion is required while retrieving Purchase Order details.

    Note You can clear this check box if WBS element conversion is not required while retrieving the Purchase Order details. This helps to avoid extra table call and ensures that the Purchase Order details are retrieved faster.
14. Optional. Select the **Enable data massaging** check box to enable data massaging before CreateInvoice and ParkInvoice operations.
15. Optional. In the **Custom wrapper BAPI for fetching payment terms** box, specify the custom wrapper BAPI name if the standard function module named MSR20_MD_PAYTERMS_GETLIST is not remote-enabled in the SAP environment.

    Note To create custom wrapper BAPI, refer the section titled Create custom wrapper BAPI for fetching payment terms.
16. Click **Save**.

Identify connectivity issues

You can identify connectivity issues with Perceptive Content, SAP, and Cache using the SAP Financial Connector Diagnostics web interface. To identify connectivity issues, complete the following steps.

1. In a browser, type the SAP Financial Connector Diagnostics address in the format `http://<Perceptive Connect Runtime machine name>:<port>/diag`.
2. In **SAP Financial Connector Diagnostics**, select the appropriate check box to detect connectivity of any of the following applications with SAP Financial Connector.
    - **Perceptive Content connectivity**
    - **ERP Connectivity : <Unique ID of the ERP/Cache as configured in Perceptive Connect Runtime Web Console>**

3. Click **Start** to run the diagnostics.

Set the default SAP connection

If you have multiple active SAP connections, you can set a default connection if you select no other value while mapping the operations. To set the default SAP connection, complete the following steps.

1. In **Perceptive Connect Runtime Web Console**, under **Perceptive Connect**, click **View Configuration**.
2. Under **General**, click the **Edit the configuration values** ✎ button next to **ERP Async Operation Config**.
    - In the **Default ERP Service ID** list, select a value from all configured SAP connections and click **Save**.
3. Similarly, under **General**, click the **Edit the configuration values** ✎ button next to **ERP Sync Operation Config**.
    - In the **Default ERP Service ID** list, select a value from all configured SAP connections and click **Save**.

Configure a cache

You can implement a cache for faster data retrieval. To implement a cache, complete the following steps.

1. In the **Perceptive Connect Runtime Web Console**, under **Perceptive Connect**, click **View Configuration** and in the **View Configuration** page, under **ERP Cache Config,** click the **Create new factory configuration** ✚ button.
2. In the **ERP Cache Config** dialog box, complete the following actions.
    1. In the **Unique ID** box, type a unique name for the cache.
    2. In the **Origin ERP id** box, type the name of the ERP connection to the cache.
    3. In the **Cache path** box, you can specify the location to store the cache database. The default path of the cache database is **ErpCache** in the **Perceptive Connect Runtime** directory.
    4. In the **Refresh cache at** box, type the time when you want to schedule refreshing of the cache. The default time is 12:00 AM.

       Note The correct time format in the **Refresh cache at** box is hh:mm AM/PM. If you enter time in an incorrect format, the cache refreshes at the default time, though the **Refresh cache at** box retains the incorrect time format. You can view the actual time when the scheduler runs from the **Log Service** screen.
    5. Select the **Replace expired data during cache refresh** check box to replace the expired data in cache with current data available in SAP during cache refresh.
    6. In the **Partial search count** box, enter the number of partial searches you want to retain after a cache refresh.
    7. In the **Detail search count** box, enter the number of detail searches you want to retain after a cache refresh.
    8. Select the **Clear cache** check box to clear the cache.

9. In the **CompanyList expiration period** box, type the duration for which the cached data is available on your computer. After the data expires, new searches retrieve live data from SAP.
10. In the **CompanyList expiration period unit** box, select the unit of the expiration period, such as days, hours, or minutes.
11. Similarly, enter the expiration duration and unit to configure a cache for the operations CompanyDetail, CostCenterDetails, CostCenters, CurrencyList, TaxCodes, Vendors, VendorDetails, VendorRemitToList, PaymentTerms, GLAccounts, ProfitCenters, PurchaseRequisition, and VendorCompanyMaster.

3. Click **Save**.

Configure duplicate invoice check

Duplicate invoice check feature enables Perceptive Connector for SAP Financials to identify duplicate invoices and credit memos.

To configure duplicate invoice check, complete the following steps.

1. In **Perceptive Connect Runtime Web Console**, under **Perceptive Connect**, click **View Configuration**.
2. In the **View Configuration** page, click **SAP Duplicate Invoice Check Configuration**.
3. In the **SAP Duplicate Invoice Check Configuration** dialog box, complete the following steps.
   1. In the **DupCheck ID** box, specify a unique ID for the duplicate check.
   2. Select the check boxes for the parameters based on which you want to perform duplicate check. For example, **DupCheck ID**.

   Note Although unlikely, duplicate invoice check using Gross Amount for non-PO invoices may not be accurate. Please refer to your SAP FI consultant to understand if this scenario may apply to your configuration.
4. Click **Save**.

   Note You must associate the DupCheck ID to the SAP instance to which you want to enable duplicate invoice check.

About data massaging

Data massaging functionality allows you to modify invoice data using custom JavaScripts before CreateInvoice and ParkInvoice operations are performed. You can edit custom JavaScripts either locally at the server or using the data massaging web interface.

Configure data massaging at the server locally

To configure data massaging at the server locally, complete the following steps.

1. Navigate to the **[*drive*:]/{*directory where Perceptive Connect Runtime is extracted*}/SAPCustomScript** directory.
2. Open the file **<*ERP instance name*>_Custom_Invoice_Operation.js**.
3. Edit the JavaScript under the **customInvoiceDataMassager** and **customAccountInvoiceDataMassager** functions to modify invoice data.

Note The customInvoiceDataMassager function accepts two parameters – invoice and logger. The input mapping XML of CreateInvoice and ParkInvoice operations includes the properties of the invoice parameter that you can use under this function.

Note Before you perform data massaging for the first time, ensure that the **<instance name>_invoice_operation.js** file displays the appropriate functions. If they do not display, manually add them.

4. Save the file.
5. Open **Perceptive Connect Runtime Web Console**.
6. On the **Perceptive Connect** menu, click **View Bundles**.
7. Locate the **ErpImplSap** bundle and click the **Refresh Package Imports** ⟳ button associated with it.

Configure data massaging using the web interface

Data massaging web interface is available when the ErpDataMassagingUIComponent-<version>.jar bundle is installed.

To configure data massaging using the web interface, complete the following steps.

1. Open a web browser and, in the address bar type, the address for data massaging web interface in the format `<IP address of the Perceptive Connect Runtime>:<port number>/sfc/dataMassaging.html`
2. In the **Select an ERP instance** drop-down, select the appropriate ERP instance.
3. Under **Edit script**, edit the JavaScript under the **customInvoiceDataMassager** and **customAccountInvoiceDataMassager** functions to modify invoice data.

   Note The customInvoiceDataMassager function accepts two parameters – invoice and logger. The input mapping XML of CreateInvoice and ParkInvoice operations includes the properties of the invoice parameter that you can use under this function.

   Note Before you perform data massaging for the first time, ensure that the custom JavaScript for the selected ERP instance displays the appropriate functions. If they do not display, manually add them.

4. Click **Save**.

Configure Perceptive Content to use the connector

Perceptive Content workflow with Envoy services facilitates back-end integration of Perceptive Content with your SAP application. The Integration automated system queue you implement in your workflow process sends a web service notification using the Envoy service operation name you define.

The application receives the data you configure, such as the eForm data. After this process is complete, your application sends a success response using Perceptive Integration Server, and Perceptive Content can then move the file forward in the Perceptive Content workflow for additional processing.

Configure ASQEndpointService

For ImageNow 6.7, Perceptive Content 7.0, and Perceptive Content 7.1

To configure your ASQEndpointService web service, complete the following steps.

1. In the **Management Console**, in the left pane, click **Envoy Services** and then click **New**.
2. In the **Envoy Services** dialog box, in the **Definition** page, set the following attributes.
    1. In the **Name** box, type ASQEndpointService.
    2. Optional. In the **Description** box, type a description.
    3. In the **URI** box, type the URI for your connector server.

        http://<*server* IP:port>/ws/workflowTrigger?wsdl
    4. In the **Authentication** list, select **None**.
    5. Optional. If you want to enable interceptor logging for the remote service, select the **Enable interceptor logging** check box.
3. Click **Next**.
4. In the **Operations** page, in the **ASQEndpointService** operation, select the check box for **InvokeTrigger** and click **Finish**.
5. Click **Finish**.

Configure ERPCommonOperationsService

For ImageNow 6.7, Perceptive Content 7.0, and Perceptive Content 7.1

To configure your ERPCommonOperationsService, complete the following steps.

1. In the **Management Console**, in the left pane, click **Envoy Services** and then click **New**.
2. In the **Envoy Services** dialog box, in the **Definition** page, set the following attributes.
    1. In the **Name** box, type `ERPCommonOperationsService`.
    2. Optional. In the **Description** box, type a description.
    3. In the **URI** box, type the URI for your connector server.

        `http://<server IP:port> /ws/ErpSyncOperations?wsdl`
    4. In the **Authentication** list, select **None**.
    5. Optional. If you want to enable interceptor logging for the remote service, select the **Enable interceptor logging** check box.
3. Click **Next**.
4. In the **Operations** page, in the **ERPCommonOperationsService** operation, select the check boxes for the following operations.
    - GetCompanyDetails
    - GetCompanyList
    - GetCostCenterDetails
    - GetCostCenters
    - GetCurrenciesForCountry
    - GetCurrencyList
    - GetEmailAddresses

- GetErpServiceList
- GetGLAccounts
- GetPaymentTermList
- GetProfitCenterList
- GetPurchaseOrderDetails
- GetPurchaseOrderList
- GetPurchaseRequisitionDetails
- GetTaxCodes
- GetVendorDetails
- GetVendorList
- GetVendorRemitToList

5. Click **Finish**.

Create the workflow process

To create a workflow process, complete the following steps.

1. In **Management Console**, in the left pane, click **Workflow**.
2. In the right pane, on the **Workflow** tab, click **New**.
3. In the **Add Process** dialog box, in the **Name** box, type a name, and then in the **Description** box, type the description of the process that will appear in the ToolTip for the process.
4. Click **OK**, select the process, and then click **Modify**.

   Step result: The **Perceptive Content Workflow Designer** opens to create queues, routes, and workflow rules.

Create the success and failure queues

You need to configure the success and failure work queues that the Integration ASQ requires to process the documents.

To create the work queue for success and failure, complete the following steps.

1. Select the workflow you created in Create the workflow process and click **Modify**.
2. In the **Workflow Designer** window, in the left pane, under **Queues**, select the **Work** queue and drag it to the right in your process diagram. Repeat this step to create a second work queue.
3. Double-click the first work queue, and in the **Queue Properties** dialog box, in the **Name** box, type **Failed**, and then click **OK**.
4. Repeat the previous step to name a queue **Success**.

Create an Integration ASQ

For ImageNow 6.7, Perceptive Content 7.0, and Perceptive Content 7.1

The ASQ integrates the workflow process with Perceptive Connect Runtime. The ASQ uses the success and failure work queues to indicate the post-processing status of the document.

To create the integration queue, complete the following steps.

1. In the **Workflow Designer** window, in the left pane, under **Queues**, select the **Integration** automated system queue (ASQ) and drag it into your process diagram to the right.
2. Double-click the integration queue and in the **Queue Properties** dialog box, in the **Name** box, type a name for the queue.
3. Under **Automated Action**, set the following attributes.
    1. Under **Success Action**, in the **Process** list select the workflow process and in the **Queue** list select **Success**.
    2. Under **Failure Action**, in the **Process** list select the workflow process and in the **Queue** list select **Failure**.
    3. Under **Envoy Service**, in the **Service Operation Name** list, click the Envoy Service name, such as **ASQEndpointService::invoke**, and click **OK**.

       Note Record the queue **ID** value to use in the channel mapping.
4. Repeat these steps for other operations, such as, **CreateInvoice**, **ParkInvoice**, **PostInvoice**, and **CancelInvoice**.

Create a Connect ASQ

For Perceptive Content 7.2.2

The ASQ integrates the workflow process with Perceptive Connect Runtime. The ASQ uses the success and failure work queues to indicate the post-processing status of the document.

To create the integration queue, complete the following steps.

1. In the **Workflow Designer** window, in the left pane, under **Queues**, select the **Connect** automated system queue (ASQ) and drag it into your process diagram to the right.
2. Double-click the **Connect** queue and in the **Queue Properties** dialog box, in the **Name** box, type a name for the queue.
3. Under **Automated Action**, set the following attributes.
    1. Under **Success Action**, in the **Process** list select the workflow process and in the **Queue** list select **Success**.
    2. Under **Failure Action**, in the **Process** list select the workflow process and in the **Queue** list select **Failure**.

       Note Record the queue **ID** value to use in the channel mapping.
4. Repeat these steps for other operations, such as, **CreateInvoice**, **ParkInvoice**, **PostInvoice**, and **CancelInvoice**.

About channel configuration

A channel is a combination of a trigger, an action, and one or more results. You create and configure channels using the Perceptive Connect Runtime Web Console. When you create a channel, you first select a trigger and then select a compatible action available with the connector, then you map the channel to write the results to Perceptive Content.

Create and configure a channel

To create and configure a channel, complete the following steps.

1. In **Perceptive Connect Runtime Dashboard**, under **Business Administrators**, click **Create a channel**. The browser redirects you to **http://<*Perceptive Connect Runtime machine*>:<*port*> /ui/#/trigger**.
2. In the **Select a trigger** list, select the **Integration ASQ Trigger**.
3. To set the queue that acts as a trigger, in the **Workflow Queue ID** box, type the **ID** value you recorded in the Create the queues topic or complete the following steps.
    1. In **Perceptive Content Management Console**, in the left pane, click **Workflow**, and on the **Workflow** tab, click **Modify**.
    2. In the **Perceptive Content Workflow Designer**, double-click the **Connect** queue to view the queue properties.

        Note For ImageNow 6.7, Perceptive Content 7.0 and Perceptive Content 7.1, double-click the **Integration** queue to view the queue properties.
    3. Under **Details**, note the **ID** value, such as 321YX5N_00004JFZG0000E6.
4. To associate an action with the trigger, in the **Select an action** list, click an action. Based on your requirement, you can select from the following actions.
    - **CancelInvoice**
    - **PostInvoice**
    - **CreateInvoice**
    - **ParkInvoice**

    Note After you select the appropriate action, a configurable XML script appears.
5. Click **Next**.
6. Under **Configure the channel results**, in the wizard box, modify the XML script that allows mapping of results to Perceptive Content document properties or indexes.
7. Click **Save Channel**.
8. In the popup that appears, click **OK** to enable the channel.

Function modules for SAP authorization

Function modules

The table in this section explains the function modules and related SAP authorization performed by each module.

Function modules	Description	Function Group	Requirement
BAPI_GL_ACC_GETLIST	This function module provides a list of G/L accounts associated with a company code. SAP determines the data and returns it to a table. Any errors that occur are provided in the form of a return code message. SAP does not convert the value of the input parameter COMPANYCODE, so leading blank characters within the value are not cleared. Based on the language you specify under the LANGUAGE parameter, descriptions appear in the specified language if translated version of the descriptions is available in that language.	FBS/3006	The same authorization checks are performed for this function module that are performed when the information is displayed within the transaction on the user interface.
CONVERSION_EXIT_PRCOR_INPUT	Format/Conversion RFC	CJPNCORE	
BAPI_COMPANYCODE_GETLIST	This function module provides you with a list of company codes. SAP provides both the keys and the names of the company codes if they exist in the system. Any errors that occur are provided in the form of a return code message. Text for company code is returned if it exists in the system.	FBASCORE/0002	The same authorization checks are performed for this function module that are performed when the information is displayed within an R/3 transaction.
BAPI_COMPANYCODE_GETDETAIL	This function module enables you to access additional information for a company code. SAP provides the detail data and address data associated with the company code. Any errors that occur are provided in the form of a return code message.	FBASCORE/0002	

Function modules	Description	Function Group	Requirement
BAPI_PO_GETDETAIL	This function module enables you to list the detailed data on a purchase order.	MEW/MEWP	The following authorization objects are verified with the activity 03 (Display). • M_BEST_BSA Document type in purchase order • M_BEST_EKG Purchasing group in purchase order • M_BEST_EKO Purchasing organization in purchase order • M_BEST_WRK Plant in purchase order
BAPI_VENDOR_GETDETAIL	Vendor Detail Information. Use this function module to call detailed data for a vendor.	VS/LC01	
MSR20_MD_PAYTERMS_GETLIST	This function module returns a list of Terms of Payment.	MBA_MSR20/MSR20_MD	
BAPI_INCOMINGINVOICE_PARK	You can use this function module to park incoming invoices. When header data is transferred, you use the parameter INVOICE_IND to determine whether SAP generates a credit memo or an invoice. If you leave the parameter initial, the system generates a credit memo. If you enter the value X, the system generates an invoice. You use this function module if information about posting an invoice document is missing in the invoice document. The balance can be any value except zero. This example lists some of the information that is updated when you use this module in case of online transaction.	MRM/MRM_BAPI	The following authorization object is checked with activity 77. M_RECH_WRKPlant authorization for incoming invoices that are entered

Function modules	Description	Function Group	Requirement
BAPI_INCOMINGINVOICE_POST	This function module enables you to post provisional invoice documents. Provisional invoice documents have the following status. • On hold • On hold (from parking) • Parked • Parked as complete • Parked and released Restrictions It is not possible to post invoice documents that are flagged for background invoice verification or that are saved within the framework of the EDI process using this module.	MRM/MRM_BAPI	The authorization object, M_RECH_WRK, is checked with activity 77 for parked invoice documents. For invoice documents with any other status, activity 01 is checked.
BAPI_INCOMINGINVOICE_CREATE	You can use this function module to create and post incoming invoices. When header data is transferred, you use the parameter INVOICE_IND to determine whether SAP generates a credit memo or an invoice. If you leave this parameter as blank, the system posts a credit memo. If you enter the value X, the system posts an invoice. If you want to enter an invoice for a returns order, you must leave the initial value in the parameter INVOICE_IND. As in case of online transaction, SAP updates the purchase order history and stock valuation data in Materials Management and creates accounting documents in Accounting.	MRM/MRM_BAPI	SAP checks the following authorization object using activity 01. • M_RECH_WRK Authorization to create invoices in certain plants
BAPI_INCOMINGINVOICE_CANCEL	With this function module, you can create a reversal document by entering the original document.	MRM/MRM_BAPI	SAP checks the following authorization object with activity 01. • M_RECH_WRK Plant Authorization for entering Invoice

Function modules	Description	Function Group	Requirement
BAPI_PROFITCENTER_GETLIST	This function module supplies a list of profit centers sorted by controlling area, and person in charge within each controlling area.	KE1A/0015	You need authorization for the activity "03" (Display) for the object "K_PCAS_PRC" (Maintain profit centers) in the current controlling area. You also need authorization for the activity "03" (Display) for the object "K_PCAS_REP" (Summary and line item reports) for the relevant profit center.
BAPI_COSTCENTER_GETLIST1	This function module identifies a list of cost centers that conform to the selection criteria in the interface. Afterwards, you can use the function module BAPI_COSTCENTER_GETDETAIL1 to determine more detailed information for a single cost center.	KBAS/0012	
BAPI_COSTCENTER_GETDETAIL1	This function module supplies detailed information for a selected cost center. You can determine a list of cost centers prior to this by using the function module BAPI_COSTCENTER_GETLIST1.	KBAS/0012	
BAPI_PR_GETDETAIL	This function module provides details for MEREQ Object	MEREQ/2105	
BAPI_TRANSACTION_COMMIT	This function module executes a COMMIT WORK command. It is required for transactions developed outside the system where SAP is installed to change SAP data by using BAPI calls. When you call BAPI functions in your program that change data in SAP, you must call this module to write the changes to the database.	SBAPI/BAPT	

Function modules	Description	Function Group	Requirement
BAPI_TRANSACTION_ROLLBACK	This function module executes a ROLLBACK WORK command. It is required for transactions developed outside the SAP system that change data in SAP by using BAPI calls. If you call BAPI functions in your program that change data in SAP, this module can prevent these changes from being written to the database, provided that these changes have not already been transferred to the database with a COMMIT WORK command.	SBAPI/BAPT	
RFC_READ_TABLE • TCURT • T500W • ADR6 • EKKO • EKPO • T007S • T005 • LFB1 • LFA1 • T001 • T042Z • LFBK • BNKA • WYT3 • TKA02 • PRPS	External access to R/3 tables via RFC.	SDTI/SDTX	

Notes

- When a function module is called remotely, an automatic authorization check is performed if the profile parameter auth/rfc_authority_check is set to 1. The authorization check uses the authorization object S_RFC to check whether the user defined in the destination has RFC authorization for the function group of the called function module.
- Authorization object S_RFC must also contain RFC function groups that help perform technical processes during communication, if any such function groups exist.

  - For tRFC and qRFC calls between two SAP systems, include function groups ERFC and ARFC.
  - For all RFC calls in external (non-SAP) systems, include function group SYST.
- The automatic authorization check is made by implicitly calling function module AUTHORITY_CHECK_RFC. If it detects an authorization issue, this function module triggers one of the exceptions defined in its interface, namely USER_DONT_EXIST or RFC_NO_AUTHORITY, thus triggering a runtime error. We recommend that you call the function module AUTHORITY_CHECK_RFC explicitly before a remote call, thus enabling you to handle any exceptions that may arise. If it detects the relevant authorization, the function module does not return an explicit result. Instead, like all function modules that are performed successfully, it sets sy-subrc to 0.

Asynchronous operations

About asynchronous operations

The asynchronous operations available with Perceptive Connector for SAP Financials enable you to call various methods through the ASQ (Automated System Queue).

ErpAsyncConnector-<version>.jar: CreateInvoice and ParkInvoice

CreateInvoice and ParkInvoice help to create and park, respectively, all types of invoices, including Purchase Order (PO) based invoices, Non-PO based invoices, and Credit Memos.

The CreateInvoice operation calls the following function module.

- BAPI_INCOMINGINVOICE_CREATE

The ParkInvoice operation calls the following function module.

- BAPI_INCOMINGINVOICE_PARK

The following table displays the activities that each of the function modules perform.

BAPI_INCOMINGINVOICE_CREATE	BAPI_INCOMINGINVOICE_PARK
You can use this module to perform the following activities. • Enter and post invoices with multiple account assignment • Change the account assignment of invoices, add account assignments, and post the invoices • Enter invoices for blanket purchase orders • Enter and post invoices for purchase orders containing planned delivery costs • Enter and post invoices containing unplanned delivery costs • Enter and post subsequent debits/credits • Use document types with external number assignment • Enter and post invoices with withholding tax • Enter and post invoice with more than one vendor line, different payment methods, and different terms of payment • Enter and post invoices for returns orders • Enter and post invoices for transportation service providers • Enter and post invoices for external service orders • Post to G/L accounts • Post to material accounts • Enter and post invoices for a one-time vendor • Enter and post invoices with a different payee	You can use this module to perform the following activities. • Park invoices with multiple account assignment • Change and park invoice account assignments • Park invoices for blanket purchase orders • Park invoices for purchase orders with planned delivery costs • Park invoices with unplanned delivery costs • Park subsequent credits and debits • Use document types with external number assignment • Park invoices with withholding tax • Park invoices with several creditor lines and different methods and terms of payment • Park invoices for purchase orders for external services • Park invoices with G/L account postings • Park invoices with material account postings • Park invoices for a one-time vendor • Park invoices with a different payee

While mapping the operations, consider the following guidelines.

- If you are mapping for PO based invoices, map InvoiceLineItemList and its child parameters. You can map InvoiceGLLineItemList and also InvoiceTaxLineItemList to PO based invoices.
- If you are mapping for non-PO based invoices, map InvoiceGLLineItemList, InvoiceTaxLineItemList, and their respective child parameters.
- You must manually map the PO1 to PO10 fields under InvoiceLineItemList in the mapping XML of the CreateInvoice and ParkInvoice operations. If mapping for PO1 to PO10 fields are not configured, you must map them to blank values or eForm fields. These mappings enable the connector to perform

customized conditional checks using a data massaging script on fields retrieved from the GetPurchaseOrderDetails operation when you perform a CreateInvoice or ParkInvoice operation.

The following XML template is used to configure the input parameters for CreateInvoice and ParkInvoice operations.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>ErpId</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>EnableDuplicateInvoiceCheck</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>InvoiceInd</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>DocType</c:name>
        <c:none/>
    </c:parameter>
    ...
    <c:parameter>
        <c:name>PostingLogicForReturn</c:name>
        <c:none/>
    </c:parameter>
    <c:rowset>
        <c:name>InvoiceLineItemList</c:name>
        <c:none/>
        <c:mapping>
            <c:parameter>
                <c:name>Match_INV</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>InvoiceDocItem_INV</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>PONumber</c:name>
                <c:none/>
            </c:parameter>
            ...
            <c:parameter>
                <c:name>RetentionDueDate</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>IrrelevantRetention</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>ValuationType</c:name>
                <c:none/>
            </c:parameter>
        </c:mapping>
    </c:rowset>
    <c:rowset>
        <c:name>InvoiceGLLineItemList</c:name>
        <c:none/>
        <c:mapping>
            <c:parameter>
                <c:name>Match_GL</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>InvoiceDocItem_GL</c:name>
                <c:none/>
            </c:parameter>
            ...
            <c:parameter>
                <c:name>ProfitSegmNumber</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>Plant</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>BudgetPeriod</c:name>
                <c:none/>
            </c:parameter>
        </c:mapping>
    </c:rowset>
    <c:rowset>
        <c:name>InvoiceTaxLineItemList</c:name>
        <c:none/>
        <c:mapping>
            <c:parameter>
                <c:name>Match_TAX</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>InvoiceDocItem_TAX</c:name>
                <c:none/>
            ...
            <c:parameter>
                <c:name>TaxBaseAmount</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>TaxJurisdiction</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>TaxJurCodeDeep</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>CondType</c:name>
                <c:none/>
            </c:parameter>
        </c:mapping>
    </c:rowset>
    <c:rowset>
        <c:name>InvoiceWithholdingTaxLineItemList</c:name>
        <c:none/>
        <c:mapping>
            <c:parameter>
                <c:name>Match_WITHHOLDING_TAX</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>InvoiceDocItem_WITHHOLDING_TAX</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>SplitKey</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>WiTaxType</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>WiTaxCode</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>WiTaxBase</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>WiTaxAmt</c:name>
                <c:none/>
            </c:parameter>
            <c:parameter>
                <c:name>WiTaxWithheldAmt</c:name>
                <c:none/>
            </c:parameter>
        </c:mapping>
    </c:rowset>
</c:inputs>
```

You can include more parameters in the input mapping XML by enabling them using the configuration file.

The following table displays the input parameters with their properties.

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	ErpId	String	SAP connection that the connector operations use. For async calls, ErpId passes as mapped operation. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.	10		Optional
Input	EnableDuplicateInvoiceCheck	Boolean	If the value is set to TRUE, the duplicate invoice check is enabled. To disable duplicate invoice check, set the value to FALSE.	1		Optional
Input	InvoiceInd	String	This is a flag to indicate the type of invoice. Possible invoice types X = Regular Invoice “ ” = Credit memo	1	INVOICE_IND	Mandatory
Input	DocType	String	Document Type. Typically RE, but you can customize a doc type. It should consist of two characters only.	2	DOC_TYPE	Optional
Input	DocDate	Date	Date when the document is created.	8	DOC_DATE	Mandatory
Input	PostingDate	Date	Date when the document is posted in SAP. In the absence of a posting date, the application takes the system date on which the invoice is posted.	8	PSTNG_DATE	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	RefDocNo	String	This is the Invoice Number. The string has a limit of 16 characters, so invoice numbers with long formats may be recorded elsewhere.	16	REF_DOC_NO	Optional
Input	CompanyCode	String	Company code/ Business unit ID	4	COMP_CODE	Mandatory
Input	DiffInvParty	String	Vendor ID	10	DIFF_INV	Optional
Input	Currency	String	Local currency for invoice generation	5	CURRENCY	Mandatory
Input	CurrencyISO	String	Currency code in ISO 4217 format	3	CURRENCY_ISO	Optional
Input	DirectExchRate	Double	Exchange rate of the currency in invoice with USD	9,5	EXCH_RATE	Optional
Input	IndirectExchRate	Double	Used within Europe for exchange rates with the Euro (EUR)	9,5	EXCH_RATE_V	Optional
Input	GrossAmount	Double	Total amount in invoice	23,4	GROSS_AMOUNT	Mandatory
Input	TaxAutoCalcInd	Boolean	Indicator for auto calculate tax applicability		CALC_TAX_IND	Optional
Input	PaymentTerm	String	Payment term applied for the invoice	4	PMNTTRMS	Optional
Input	BaselineDate	Date	Date from which the system can apply the periods of the terms of payment to calculate cash discount amounts and the due date of an invoice	8	BLINE_DATE	Optional
Input	CashDiscDays1	Double	Cash discount days 1	3	DSCT_DAYS1	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	CashDiscDays2	Double	Cash discount days 2	3	DSCT_DAYS2	Optional
Input	NetPaymentTermPeriod	Double	Net payment term period	3	NETTERMS	Optional
Input	CashDiscPerc1	Double	Cash discount percentage 1	5,3	DSCT_PCT1	Optional
Input	CashDiscPerc2	Double	Cash discount percentage 2	5,3	DSCT_PCT2	Optional
Input	IVCategory	String	Source of a logistics invoice verification document	1	IV_CATEGORY	Optional
Input	HeaderText	String	Document header text	25	HEADER_TXT	Optional
Input	PaymentBlock	String	Payment block key	1	PMNT_BLOCK	Optional
Input	DeliveryCosts	Double	Unplanned delivery costs	23,4	DEL_COSTS	Optional
Input	DeliveryCostTaxCode	String	Tax code for delivery costs	2	DEL_COSTS_TAXC	Optional
Input	DeliveryJurisdictionCode	String	Jurisdiction code for delivery costs	15	DEL_COSTS_TAXJ	Optional
Input	ExtUserName	String	Entered by external system user	12	PERSON_EXT	Optional
Input	PaymentMethod	String	Payment method	1	PYMT_METH	Optional
Input	PaymentMethodSupplement	String	Payment method supplement	2	PMTMTHSUPL	Optional
Input	InvDocNo	String	Document number of an invoice document	10	INV_DOC_NO	ParkInvoice – Mandatory CreateInvoice – Optional
Input	SCBankIndicator	String	State central bank indicator	3	SCBANK_IND	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	SupplyingCountry	String	Supplying country	3	SUPCOUNTRY	Optional
Input	ServiceIndicator	String	Service indicator for foreign payment	1	BLLSRV_IND	Optional
Input	RefDocNoLong	String	Reference document number	35	REF_DOC_NO_LONG	Optional
Input	CashDiscAmt	Double	Cash discount amount in document currency	23,4	DSCT_AMOUNT	Optional
Input	POSubscriberNo	String	ISR Subscriber number	11	PO_SUB_NO	Optional
Input	POCheckDigit	String	ISR check digit	2	PO_CHECKDG	Optional
Input	POReferenceNo	String	ISR reference number	27	PO_REF_NO	Optional
Input	Payee	String	Payee/payer	10	PAYEE_PAYER	Optional
Input	PartnerBankType	String	Partner bank type	4	PARTNER_BK	Optional
Input	HouseBankId	String	Short key for a house bank	5	HOUSEBANKID	Optional
Input	AssignmentNo	String	Assignment number	18	ALLOC_NMBR	Optional
Input	PaymentReference	String	Payment reference	30	PAYMT_REF	Optional
Input	InvRefNo	String	Document number for invoice reference	10	INV_REF_NO	Optional
Input	InvYear	String	Fiscal year of the relevant invoice (for credit memo)	4	INV_YEAR	Optional
Input	InvRecDate	Date	Invoice receipt date	8	INV_REC_DATE	Optional
Input	PlanningLevel	String	Planning level	2	PLANNING_LEVEL	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	PlanningDate	Date	Planning date	8	PLANNING_DATE	Optional
Input	FixedPaymentTerm	String	Fixed payment terms	1	FIXEDTERMS	Optional
Input	BusinessLocation	String	Business area	4	BUS_AREA	Optional
Input	LotNumber	String	Lot Number for Documents	10	LOT_NUMBER	Optional
Input	ItemText	String	Item text	50	ITEM_TEXT	Optional
Input	FiscalType	String	Fiscal type	2	J_1BNFTYPE	Optional
Input	EUTriangularDeal	String	Indicates a triangular deal within the EU Delivery of goods as part of a triangular deal must be identified when entering such information in the line item and must be listed separately in the EU sales list.	1	EU_TRIANG_DEAL	Optional
Input	ReportingCountry	String	Reporting country for delivery of goods within the EU	3	REPCOUNTRY	Optional
Input	VATRegNo	String	For every EU company that carries out transactions with other EU member states, you should specify the VAT registration number on invoices and purchase orders.	20	VAT_REG_NO	Optional
Input	BusinessPlace	String	Business place	4	BUSINESS_PLACE	Optional
Input	TaxExchRate	Double	Exchange rate for taxes	9	TAX_EXCH_RATE	Optional
Input	GoodsAffectedIndicator	String	Indicator for capital goods affected	1	GOODS_AFFECTED	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input	RetDueDate	Date	Due date for retention	8	RET_DUE_PROP	Optional
Input	PostingLogicForDelivery	String	Invoice / credit memo posting logic for delivery items	1	DELIV_POSTING	Optional
Input	PostingLogicForReturn	String	Invoice / credit memo posting logic for return items	1	RETURN_POSTING	Optional
Input	InvoiceLineItemList	Array of rows	This array of parameters is used for PO invoices only			
Input – Rowset	Match_INV	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1		Optional
Input – Rowset	InvoiceDocItem_INV	Integer	Document item in invoice document. If this value is set to 0, the application auto-populates an incremental number while posting to SAP.	6	INVOICE_DOC_ITEM	Optional
Input – Rowset	PONumber	String	Purchase order number	10	PO_NUMBER	Mandatory
Input – Rowset	POItem	Long	Item number of purchasing document	5	PO_ITEM	Mandatory
Input – Rowset	RefDocNo_INV	String	Document number of a reference document. This is the Invoice Number. The string has a limit of 16 characters, so invoice numbers with long formats may be recorded elsewhere.	10	REF_DOC	Optional
Input – Rowset	RefDocYear	Integer	Fiscal year of current period	4	REF_DOC_YEAR	ParkInvoice – Mandatory CreateInvoice – Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	RefDocItem	Integer	Item of a reference document	4	REF_DOC_IT	ParkInvoice – Mandatory CreateInvoice – Optional
Input – Rowset	DebitCreditInd_INV	String	Indicator for subsequent debit / credit	1	DE_CRE_IND	Optional
Input – Rowset	TaxCode	String	Tax code	2	TAX_CODE	ParkInvoice – Optional CreateInvoice – Mandatory
Input – Rowset	TaxJurCode	String	Tax jurisdiction	15	TAXJURCODE	ParkInvoice – Optional CreateInvoice – Mandatory
Input – Rowset	ItemAmount	Double	Amount in document currency	23,4	ITEM_AMOUNT	Mandatory
Input – Rowset	Quantity	Long	Quantity of the line item in a purchase order	13,3	QUANTITY	Mandatory
Input – Rowset	UOM	String	Unit of measure for purchase order	3	PO_UNIT	Mandatory
Input – Rowset	InternalUOM	String	Internal unit of measure for purchase order in SAP	10	PO_UNIT	Optional
Input – Rowset	UnitISO	String	ISO codes for standard measurement units are used for data exchange via Electronic Data Interchange (EDI) with external systems. To implement communication with external systems, you must assign ISO codes to SAP internal UOM.	3	PO_UNIT_ISO	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	POPriceQuantity	Double	Quantity in purchase order for price unit	13,3	PO_PR_QNT	Mandatory
Input – Rowset	POPriceUOM	String	Order price unit for purchasing	3	PO_PR_UOM	Optional
Input – Rowset	POPriceUomISO	String	Purchase order price unit in ISO code	3	PO_PR_UOM_ISO	Optional
Input – Rowset	ConditionType	String	Condition type	4	COND_TYPE	ParkInvoice – Mandatory CreateInvoice – Optional
Input – Rowset	ConditionStepNo	Integer	Step number	3	COND_ST_NO	Optional
Input – Rowset	ConditionCount	Integer	Condition counter	2	COND_COUNT	ParkInvoice – Mandatory CreateInvoice – Optional
Input – Rowset	Sheetno	Integer	Entry sheet number	10	SHEET_NO	Optional
Input – Rowset	ItemText	String	Item text	50	ITEM_TEXT	Optional
Input – Rowset	FinalInvoiceIndicator	String	Indicates that the last invoice from a vendor has been received for an item and no further invoice is expected	1	FINAL_INV	Optional
Input – Rowset	LineNo	Long	Line number	10	SHEET_ITEM	ParkInvoice – Mandatory CreateInvoice – Optional
Input – Rowset	GRIRClearingInd	String	Clearing indicator for GR/IR posting for external services	1	GRIR_CLEAR_SRV	Optional
Input – Rowset	VendorFreight	String	Account number of vendor or creditor	10	FREIGHT_VEN	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	CashDiscInd	String	Indicator for line item not liable to cash discount	1	CSHDIS_IND	Optional
Input – Rowset	RetentionAmount	Double	Retention amount in document currency	13,2	RETENTION_DOCU_CURRENCY	Optional
Input – Rowset	RetentionPercentage	Double	Retention in percent	5,2	RETENTION_PERCENTAGE	Optional
Input – Rowset	RetentionDueDate	Date	Due date for retention	8	RETENTION_DUE_DATE	Optional
Input – Rowset	IrrelevantRetention	String	Item not relevant to retention	1	NO_RETENTION	Optional
Input – Rowset	ValuationType	String	Valuation types are present for each valuation category	10	VALUATION_TYPE	Optional
Input – Rowset	ItemCategory	String	Invoice line item category. This value is obtained from GetPurchaseOrderDetails lookup.			
Input – Parameter	InvoiceGLLineItemList	Array of rows	This array of parameters is used for Non-PO invoices only			
Input – Rowset	Match_GL	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1		Optional
Input – Rowset	InvoiceDocItem_GL	Numeric	Document item in invoice document	6	INVOICE_DOC_ITEM	Mandatory
Input – Rowset	GLAccount	String	General ledger account	10	GL_ACCOUNT	Mandatory
Input – Rowset	ItemAmount	Double	Amount in document currency	23,4	ITEM_AMOUNT	Mandatory

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	DebitCreditInd_GL	String	Indicates debit or credit	1	DB_CR_IND	Mandatory
Input – Rowset	NegPosting	String	Negative posting indicator	1	NEG_POSTNG	Optional
Input – Rowset	CompanyCode	String	Company code/ business unit ID	4	COMP_CODE	Mandatory
Input – Rowset	TaxCode	String	Tax on sales/purchases code	2	TAX_CODE	Mandatory
Input – Rowset	TaxJurCode	String	Tax jurisdiction	15	TAXJURCODE	Mandatory
Input – Rowset	ItemText	String	Item text	50	ITEM_TEXT	Optional
Input – Rowset	CostCenter	String	Cost center	10	COSTCENTER	Optional
Input – Rowset	SDDoc	String	Sales and distribution document number	10	SD_DOC	Optional
Input – Rowset	SalesDocItem	Numeric	Sales document item	6	SDOC_ITEM	Optional
Input – Rowset	Orderid	String	Order number	12	ORDERID	Optional
Input – Rowset	RefDate	Date	Reference date for settlement	8	REF_DATE	Optional
Input – Rowset	CommitmentItem	String	Commitment item	14	CMMT_ITEM	Optional
Input – Rowset	FundsCenter	String	Funds center	16	FUNDS_CTR	Optional
Input – Rowset	FunctionalArea	String	Functional area	4	FUNC_AREA	Optional
Input – Rowset	ProfitCenter	String	Profit center	10	PROFIT_CTR	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	Fund	String	Funds balance provided for a specific purpose by a sponsor and managed separately. Funds are used to represent funds from secondary sources.	10	FUND	Optional
Input – Rowset	BusArea	String	Business area	4	BUS_AREA	Optional
Input – Rowset	TradingPartBussArea	String	Business area of trading partner	4	TR_PART_BA	Optional
Input – Rowset	Costobject	String	Cost object	12	COSTOBJECT	Optional
Input – Rowset	Network	String	Network number for account assignment	12	NETWORK	Optional
Input – Rowset	Activity	String	Operation/activity number	4	ACTIVITY	Optional
Input – Rowset	WbsElement	String	Work breakdown structure element (WBS element)	24	WBS_ELEM	Optional
Input – Rowset	ActivityType	String	Activity type	6	ACTTYPE	Optional
Input – Rowset	RealEstateKey	String	Internal key for real estate object	8	RL_EST_KEY	Optional
Input – Rowset	PersonnelNumber	Numeric	Personnel number	8	PERSON_NO	Optional
Input – Rowset	CoBusProc	String	Business process	12	CO_BUSPROC	Optional
Input – Rowset	GrantNbr	String	Grant	20	GRANT_NBR	Optional
Input – Rowset	CommitmentItemLong	String	Commitment item	24	CMMT_ITEM_LONG	Optional
Input – Rowset	FuncAreaLong	String	Functional area	16	FUNC_AREA_LONG	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	Quantity	Double	Quantity of the line item in a purchase order	13,3	QUANTITY	Mandatory
Input – Rowset	BaseUOM	String	Base unit of measure	3	BASE_UOM	Mandatory
Input – Rowset	AllocNumber	String	Assignment number	18	ALLOC_NMBR	Optional
Input – Rowset	CashDiscIndicator	String	Cash discount indicator	1	CSHDIS_IND	Optional
Input – Rowset	TaxBaseAmount	Double	Tax base amount in document currency	23,4	TAX_BASE_AMOUNT	Optional
Input – Rowset	ProfitSegmNumber	Numeric	Profitability segment number (CO-PA)	10	PROFIT_SEGM_NO	Optional
Input – Rowset	Plant	String	An organizational unit serving to subdivide an enterprise according to production, procurement, maintenance, and materials planning aspects. It is a place where either materials are produced or goods and services provided.	4	PLANT	Optional
Input – Rowset	BudgetPeriod	String	Financial management - period of available budget authority	10	BUDGET_PERIOD	Optional

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Parameter	InvoiceTaxLineItemList	Array of rows	This array of parameters is used for PO and non-PO invoices. To enable InvoiceTaxLineItemList, set the value of TaxAutoCalcInd parameter to FALSE or leave it as blank. If the value of TaxAutoCalcInd parameter is set to TRUE, the InvoiceTaxLineItemList is ignored.			
Input – Rowset	Match_TAX	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1		Optional
Input – Rowset	InvoiceDocItem_TAX	Numeric	Document item in invoice document	6	INVOICE_DOC_ITEM	Optional
Input – Rowset	TaxCode	String	Tax on sales or purchases code	2	TAX_CODE	Mandatory
Input – Rowset	TaxAmount	Double	Tax amount in document currency	23	TAX_AMOUNT	Mandatory
Input – Rowset	TaxBaseAmount	Double	Tax base amount in document currency	32	TAX_BASE_AMOUNT	Mandatory
Input – Rowset	TaxJurisdiction	String	Tax jurisdiction	15	TAXJURCODE	Optional
Input – Rowset	TaxJurCodeDeep	String	Tax Jurisdiction Code - Jurisdiction for Lowest Level Tax	15	TAXJURCODE_DEEP	
Input – Rowset	CondType	String	Condition Type	4	COND_TYPE	
Input – Rowset	InvoiceWithholdingTaxLineItemList	Array of rows	This array of parameters is used for PO and non-PO invoices.			

Parameter type	Parameter name	Data type	Description	Field length	SAP parameter name	Requirement
Input – Rowset	Match_WITHHOLDING_TAX	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.			
Input – Rowset	InvoiceDocItem_WITHHOLDING_TAX	Numeric	Document item for withholding tax in invoice document			
Input – Rowset	SplitKey	String	Grouping key for splitting vendor line item	6	SPLIT_KEY	Optional
Input – Rowset	WiTaxType	String	Indicator for withholding tax type	2	WI_TAX_TYPE	Mandatory
Input – Rowset	WiTaxCode	String	Withholding tax code	2	WI_TAX_CODE	Mandatory
Input – Rowset	WiTaxBase	Double	Withholding tax base Amount	23	WI_TAX_BASE	Mandatory
Input – Rowset	WiTaxAmt	Double	Withholding tax amount (in document currency)	23	WI_TAX_AMT	Optional
Input – Rowset	WiTaxWithheldAmt	Double	Withheld withholding tax amount (in document currency)	23	WI_TAX_WITHHELD_AMT	Optional

The following XML template is used to configure the output parameters for CreateInvoice and ParkInvoice operations.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:outputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>InvoiceDocNumber</c:name>
        <c:noTarget/>
    </c:parameter>
    <c:parameter>
        <c:name>FiscalYear</c:name>
        <c:noTarget/>
    </c:parameter>
    <c:parameter>
        <c:name>IsDuplicateInvoice</c:name>
        <c:noTarget/>
    </c:parameter>
</c:outputs>
```

The table below lists the output parameters in the XML file.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Output	InvoiceDocNumber	String	Invoice document number	10		
Output	FiscalYear	String	Fiscal year	4		
Output	IsDuplicateInvoice	Boolean	Duplicate invoice indicator is set to TRUE when soft stop is enabled and a duplicate invoice is detected based on your configuration.			

ErpAsyncConnector-<version>.jar: PostInvoice

PostInvoice helps to post the parked invoices in SAP.

This operation calls the following function module.

- BAPI_INCOMINGINVOICE_POST

The following XML template enables you to configure the input parameters for PostInvoice operation.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>ErpId</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>InvoiceDocNumber</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>FiscalYear</c:name>
        <c:none/>
    </c:parameter>
</c:inputs>
```

The following table displays the input parameters with their properties.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement

Input	ErpId		SAP connection that the connector operations use. For async calls, ErpId passes as mapped operation. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.	10		Optional
Input	InvoiceDocNumber	String	Invoice document number	10	INV_DOC_NO	Mandatory
Input	FiscalYear	Integer	Fiscal year	4	FISC_YEAR	Optional

The following XML template enables you to configure the output parameters for PostInvoice operation.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:outputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>InvoiceDocNumber</c:name>
        <c:noTarget/>
    </c:parameter>
    <c:parameter>
        <c:name>FiscalYear</c:name>
        <c:noTarget/>
    </c:parameter>
</c:outputs>
```

The following table displays the output parameters with their properties.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Output	InvoiceDocNumber	String	Invoice document number	10		
Output	FiscalYear	Integer	Fiscal year	4		

ErpAsyncConnector-<version>.jar: CancelInvoice

CancelInvoice cancels the invoices in SAP. This operation calls the following function module.

- BAPI_INCOMINGINVOICE_CANCEL

The following XML template enables you to configure the input parameters for CancelInvoice.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>ErpId</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>InvoiceDocNumber</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>FiscalYear</c:name>
        <c:none/>
    </c:parameter>
    <c:parameter>
        <c:name>ReasonForCancel</c:name>
        <c:none/>
    </c:parameter>
</c:inputs>
```

The following table displays the input parameters of the XML file with their properties.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	ErpId		SAP connection that the connector operations use. For async calls, ErpId passes as mapped operation. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.	10		Optional
Input	InvoiceDocNumber	String	Invoice document number	10	INV_DOC_NO	Mandatory
Input	FiscalYear	Integer	Fiscal year	4	FISC_YEAR	Mandatory
Input	ReasonForCancel	String	Reason for cancelling the invoice	2	REASON_REV	Mandatory

The following XML template enables you to configure the output parameters for CancelInvoice operation.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:outputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
```

```
    <c:parameter>
        <c:name>InvoiceCancellationDocNumber</c:name>
        <c:noTarget/>
    </c:parameter>
    <c:parameter>
        <c:name>InvoiceCancellationFiscalYear</c:name>
        <c:noTarget/>
    </c:parameter>
</c:outputs>
```

The following table displays the output parameters of the XML file with their properties.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Output	InvoiceCancellationDocNumber	Integer	Document number of the cancelled invoice	10		
Output	InvoiceCancellationFiscalYear		Fiscal year of the cancelled invoice	4		

ErpAsyncConnector-<version>.jar: CheckStatus

CheckStatus returns the status of the last operation executed on a document in Perceptive Content. This is a synchronous SOAP call.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:stat="http://status.asqoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <stat:CheckStatus>
         <INDocumentId></INDocumentId>
      </stat:CheckStatus>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameter and associated properties of stat:CheckStatus.

Parameter name	Data type	Description	Field length	Requirement
INDocumentId	String	Document ID in Perceptive Content	22	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
```

```
      <ns1:CheckStatusResponse
xmlns:ns1="http://status.asqoperation.erp.pif.apef.perceptivesoftware.com/">
         <DocumentProcessStatus>
            <INDocumentId></INDocumentId>
            <OperationName></OperationName>
            <TimeStamp></TimeStamp>
            <MessageInfoList>
               <MessageCode></MessageCode>
               <Message></Message>
            </MessageInfoList>
         </DocumentProcessStatus>
      </ns1:CheckStatusResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters and associated properties of DocumentProcessStatus.

Parameter name	Data type	Description
INDocumentId	String	Document ID in Perceptive Content
OperationName	String	Last executed async operation name on the given document ID
TimeStamp	DateTime	Last executed time stamp
MessageInfoList	Array	Message information list (Refer to the table below for parameters under this object)

The following table describes the parameters and associated properties of MessageInfoList.

Parameter name	Data type	Description
MessageCode	String	Message code mostly from SAP
Message	String	Message of the above message code

ErpAsyncConnector-<version>.jar: CreateNonPOInvoice

CreateNonPOInvoice creates non-PO invoices in SAP.

This operation calls the following function module.

- BAPI_ACC_INVOICE_RECEIPT_POST

The following XML template enables you to configure the input parameters for CreateNonPOInvoice.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<c:inputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
  <c:parameter>
    <c:name>ErpId</c:name>
    <c:none/>
  </c:parameter>
  <c:parameter>
    <c:name>EnableDuplicateInvoiceCheck</c:name>
    <c:none/>
  </c:parameter>
  <c:parameter>
    <c:name>HeaderData_ObjectType</c:name>
    <c:none/>
  </c:parameter>
  ...
  <c:parameter>
    <c:name>OnetimeCustomerData_LanguageISO</c:name>
    <c:none/>
  </c:parameter>
  <c:parameter>
    <c:name>OnetimeCustomerData_IBAN</c:name>
    <c:none/>
  </c:parameter>
  <c:rowset>
    <c:name>AccountPayableLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_AP</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>AccDocLineItemNo_AP</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>PPAExcludeInd</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>GLLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_GL</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>BudgetPeriod</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>PPAExcludeInd</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>TaxDataLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
```

```
        <c:name>Match_TAX</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>NegetivePostingInd</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>CurrencyLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_CURR</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>AccDocLineItemNo_CURR</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>DiscountBase</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>IndirectQuotedExchangeRate</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>POLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>ReferenceTransaction</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>BudgetPeriod</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>POType</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>PurchaseAmountList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>ReferenceTransaction</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>ReferenceKey</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>IndirectQuotedExchangeRate</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>ScheduleLineNo</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>COPACriteriaList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_COPACriteria</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>AccDocLineItemNo_COPACriteria</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>FieldName</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>Characteristic</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>COPAValueFieldLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_COPAValue</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>AccDocLineItemNo_COPAVal</c:name>
        <c:none/>
      </c:parameter>
      ...
      <c:parameter>
        <c:name>BaseUomISO</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>QuantitiesValueField</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
  <c:rowset>
    <c:name>ExtensionLineList</c:name>
    <c:none/>
    <c:mapping>
      <c:parameter>
        <c:name>Match_EXTENSIONIN</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>Field1</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>Field2</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>Field3</c:name>
        <c:none/>
      </c:parameter>
      <c:parameter>
        <c:name>Field4</c:name>
        <c:none/>
      </c:parameter>
    </c:mapping>
  </c:rowset>
</c:inputs>
```

The table given below displays the input parameters of the XML file.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	ErpId		SAP connection that the connector operations use. For async calls, ErpId passes as mapped operation. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.	10		Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	EnableDuplicateInvoiceCheck	Boolean	If the value is set to TRUE, the duplicate invoice check is enabled at invoice level. To disable duplicate invoice check, set the value to FALSE. Default value is FALSE.	1		Optional
Input	HeaderData_ObjectType	String	Reference transaction	5	OBJ_TYPE	Optional
Input	HeaderData_ObjectKey	String	Reference key	20	OBJ_KEY	Optional
Input	HeaderData_ObjectSystem	String	Logical system of source document	10	OBJ_SYS	Optional
Input	HeaderData_UserName	String	User name	12	USERNAME	Mandatory
Input	HeaderData_HeaderText	String	Document header text	25	HEADER_TXT	Optional
Input	HeaderData_CancelObjectKey	String	Cancel: object key (AWREF_REV and AWORG_REV)	20	OBJ_KEY_R	Optional
Input	HeaderData_CompanyCode	String	Company Code	4	COMP_CODE	Mandatory
Input	HeaderData_CurrencyIso	String	CurrencyIso in the header. If the CurrencyIso in the header and in PO line items are different, CurrencyIso in PO line items take precedence.			

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	HeaderData_VendorNo	String	Vendor number in the eForm header. Vendor number in the header replaces any other vendor numbers associated with the PO line items. If there is no vendor number in the header, you can associate different vendor numbers to the PO line items.	10		
Input	HeaderData_AccountingDocumentNo	String	Accounting document number	10	AC_DOC_NO	Optional
Input	HeaderData_FiscalYear	String	Fiscal year	4	FISC_YEAR	Optional
Input	HeaderData_DocDate	Date	Document date in document	8	DOC_DATE	Optional
Input	HeaderData_PostingDate	Date	Posting date in the document	8	PSTNG_DATE	Mandatory
Input	HeaderData_TranslationDate	String	Translation date	8	TRANS_DATE	Optional
Input	HeaderData_FiscalPeriod	String	Fiscal period	2	FIS_PERIOD	Optional
Input	HeaderData_DocumentType	String	Document type	2	DOC_TYPE	Optional
Input	HeaderData_RefDocNo	String	Reference document number	16	REF_DOC_NO	Optional
Input	HeaderData_ACCInterfaceComponent	String	Component in ACC interface	4	COMPO_ACC	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	HeaderData_ReversalReason	String	Reason for reversal	2	REASON_REV	Optional
Input	HeaderData_RefDocNoLong	String	Reference document number (for dependencies see Long Text)	35	REF_DOC_NO_LONG	Optional
Input	HeaderData_AccountingPrinciple	String	Accounting principle	4	ACC_PRINCIPLE	Optional
Input	HeaderData_InvoiceReceiptDate	Date	Invoice receipt date	8	INVOICE_REC_DATE	Optional
Input	OnetimeCustomerData_Name1	String	Name 1	35	NAME	Optional
Input	OnetimeCustomerData_Name2	String	Name 2	35	NAME_2	Optional
Input	OnetimeCustomerData_Name3	String	Name 3	35	NAME_3	Optional
Input	OnetimeCustomerData_Name4	String	Name 4	35	NAME_4	Optional
Input	OnetimeCustomerData_PostalCode	String	Postal code	10	POSTL_CODE	Optional
Input	OnetimeCustomerData_City	String	City	35	CITY	Optional
Input	OnetimeCustomerData_Country	String	Country key	3	COUNTRY	Optional
Input	OnetimeCustomerData_CountryISO	String	Country key in ISO code	2	COUNTRY_ISO	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	OnetimeCustomerData_Street	String	House number and street	35	STREET	Optional
Input	OnetimeCustomerData_POBox	String	PO Box	10	PO_BOX	Optional
Input	OnetimeCustomerData_POBoxPostalCode	String	P.O. Box postal code	10	POBX_PCD	Optional
Input	OnetimeCustomerData_BankAccNoAtPostOffice	String	Account number of bank account at Post Office	16	POBK_CURAC	Optional
Input	OnetimeCustomerData_BankAccNo	String	Bank account number	18	BANK_ACCT	Optional
Input	OnetimeCustomerData_BankNo	String	Bank number	15	BANK_NO	Optional
Input	OnetimeCustomerData_BankCountry	String	Bank country key	3	BANK_CTRY	Optional
Input	OnetimeCustomerData_BankCountryISO	String	Bank country key in ISO code	2	BANK_CTRY_ISO	Optional
Input	OnetimeCustomerData_TaxNumber1	String	Tax Number 1	16	TAX_NO_1	Optional
Input	OnetimeCustomerData_TaxNumber2	String	Tax Number 2	11	TAX_NO_2	Optional
Input	OnetimeCustomerData_Tax	String	Liable for VAT	1	TAX	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input	OnetimeCustomerData_TaxEqualizationInd	String	Indicator: Business Partner Subject to Equalization Tax	1	EQUAL_TAX	Optional
Input	OnetimeCustomerData_Region	String	Region (State, Province, County)	3	REGION	Optional
Input	OnetimeCustomerData_BankControlKey	String	Bank Control Key	2	CTRL_KEY	Optional
Input	OnetimeCustomerData_DataMediumExchangeInstructionkey	String	Instruction key for data medium exchange	2	INSTR_KEY	Optional
Input	OnetimeCustomerData_DataMediumExchangeInd	String	Indicator for Data Medium Exchange	1	DME_IND	Optional
Input	OnetimeCustomerData_LanguageISO	String	Language according to ISO 639	2	LANGU_ISO	Optional
Input	OnetimeCustomerData_IBAN	String	International Bank Account Number	34	IBAN	Optional
Input – Parameter	AccountPayableLineList	Array of rows	Array of Account Payable lines or Vendor Line Items			Mandatory
Input – Rowset	Match_AP	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.			Optional
Input – Rowset	AccDocLineItemNo_AP	Integer	Accounting document line item number	10	ITEMNO_ACC	Mandatory
Input – Rowset	VendorNo	String	Account number of vendor or creditor	10	VENDOR_NO	Mandatory

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	BusinessPartnerRefKey1	String	Business partner reference key	12	REF_KEY_1	Optional
Input – Rowset	BusinessPartnerRefKey2	String	Business partner reference key	12	REF_KEY_2	Optional
Input – Rowset	LineItemRefKey	String	Reference key for line item	20	REF_KEY_3	Optional
Input – Rowset	GLAccountNo	String	General ledger account	10	GL_ACCOUNT	Optional
Input – Rowset	PaymentTerm	String	Terms of payment key	4	PMNTTRMS	Optional
Input – Rowset	BaseLineDate	String	Baseline date for due date calculation	8	BLINE_DATE	Optional
Input – Rowset	CashDiscountDays1	String	Cash discount days 1	3	DSCT_DAYS1	Optional
Input – Rowset	CashDiscountDays2	String	Cash discount days 2	3	DSCT_DAYS2	Optional
Input – Rowset	NetPaymentTermsPeriod	String	Net payment terms period	3	NETTERMS	Optional
Input – Rowset	CashDiscountPercent1	String	Cash discount percentage 1	5	DSCT_PCT1	Optional
Input – Rowset	CashDiscountPercent2	String	Cash discount percentage 2	5	DSCT_PCT2	Optional
Input – Rowset	PaymentMethodSupplement	String	Payment method supplement	2	PMTMTHSUPL	Optional
Input – Rowset	PaymentMethod	String	Payment method	1	PYMT_METH	Optional
Input – Rowset	PaymentBlockKey	String	Payment block key	1	PMNT_BLOCK	Optional
Input – Rowset	StateCentralBankInd	String	State central bank indicator	3	SCBANK_IND	Optional
Input – Rowset	SupplyingCountry	String	Supplying country	3	SUPCOUNTRY	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	SupplyingCountryISO	String	Supplier country ISO code	2	SUPCOUNTRY_ISO	Optional
Input – Rowset	ServiceInd	String	Service indicator (Foreign payment)	1	BLLSRV_IND	Optional
Input – Rowset	AssignmentNo	String	Assignment number	18	ALLOC_NMBR	Optional
Input – Rowset	ItemText	String	Item text	50	ITEM_TEXT	Optional
Input – Rowset	ISRSubscriberNo	String	ISR subscriber number	11	PO_SUB_NO	Optional
Input – Rowset	ISRCheckDigit	String	ISR check digit	2	PO_CHECKDG	Optional
Input – Rowset	ISRRefNo	String	ISR reference number	27	PO_REF_NO	Optional
Input – Rowset	WithholdingTaxCode	String	Withholding tax code	2	W_TAX_CODE	Optional
Input – Rowset	BusinessPlace	String	Business place	4	BUSINESSPLACE	Optional
Input – Rowset	SectionCode	String	Section code	4	SECTIONCODE	Optional
Input – Rowset	InstructionKey1	String	Instruction key 1	2	INSTR1	Optional
Input – Rowset	InstructionKey2	String	Instruction key 2	2	INSTR2	Optional
Input – Rowset	InstructionKey3	String	Instruction key 3	2	INSTR3	Optional
Input – Rowset	InstructionKey4	String	Instruction key 4	2	INSTR4	Optional
Input – Rowset	PPAExcludeInd	String	PPA Exclude Indicator	1	PPA_EX_IND	Optional
Input – Parameter	GLLineList	Array of rows	This array of parameters is used for Non-PO invoices.			Mandatory

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	Match_GL	Boolean	If the value is set to TRUE, the corresponding row is included for non-PO invoice creation.			Optional
Input – Rowset	AccDocLineItemNo_GL	Integer	Accounting document line item number	10	ITEMNO_ACC	Mandatory
Input – Rowset	GLAccount	String	General ledger account	10	GL_ACCOUNT	Mandatory
Input – Rowset	CompanyCode	String	Company code	4	COMP_CODE	Optional
Input – Rowset	DocPostingDate	Date	Posting date in the document	8	PSTNG_DATE	Optional
Input – Rowset	DocType	String	Document type	2	DOC_TYPE	Optional
Input – Rowset	AccountingDocNo	String	Accounting document number	10	AC_DOC_NO	Optional
Input – Rowset	FiscalYear	String	Fiscal year	4	FISC_YEAR	Optional
Input – Rowset	FiscalPeriod	String	Fiscal period	2	FIS_PERIOD	Optional
Input – Rowset	StatisticalLineItemsInd	String	Indicator for Statistical Line Items	1	STAT_CON	Optional
Input – Rowset	BusinessPartnerRefKey1	String	Business partner reference key	12	REF_KEY_1	Optional
Input – Rowset	BusinessPartnerRefKey2	String	Business partner reference key	12	REF_KEY_2	Optional
Input – Rowset	LineItemRefKey	String	Reference key for line item	20	REF_KEY_3	Optional
Input – Rowset	TaxCode	String	Tax on sales/purchases code	2	TAX_CODE	Mandatory
Input – Rowset	Transaction Key	String	Transaction key	3	ACCT_KEY	Optional
Input – Rowset	TaxJurisdiction	String	Tax jurisdiction	15	TAXJURCODE	Mandatory

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	NoCashDiscountInd	String	Indicator: Line item not liable to cash discount	1	CSHDIS_IND	Optional
Input – Rowset	AccountType	String	Account type	1	ACCT_TYPE	Optional
Input – Rowset	Assignment Number	String	Assignment number	18	ALLOC_NMBR	Optional
Input – Rowset	ItemText	String	Item text	50	ITEM_TEXT	Optional
Input – Rowset	BusinessArea	String	Business area	4	BUS_AREA	Optional
Input – Rowset	CostCenter	String	Cost Center	10	COSTCENTER	Optional
Input – Rowset	POPriceQuantity	String	Quantity in Purchase Order Price Unit	13	PO_PR_QNT	Optional
Input – Rowset	POPriceUom	String	Order Price Unit (Purchasing)	3	PO_PR_UOM	Optional
Input – Rowset	InternalPOPriceUom	String	The value for this parameter passes to SAP if value for POPriceUom is Null or Empty	3		Optional
Input – Rowset	POPriceUomISO	String	Purchase order price unit in ISO code	3	PO_PR_UOM_ISO	Optional
Input – Rowset	OrderId	String	Order number	12	ORDERID	Optional
Input – Rowset	AssetNo	String	Main asset number	12	ASSET_NO	Optional
Input – Rowset	AssetSubNo	String	Asset Subnumber	4	SUB_NUMBER	Optional
Input – Rowset	AssetValue Date	String	Asset value date	8	ASVAL_DATE	Optional
Input – Rowset	MaterialNo	String	Material number	18	MATERIAL	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	Quantity	String	Quantity	13	QUANTITY	Optional
Input – Rowset	BaseUom	String	Base unit of measure	3	BASE_UOM	Optional
Input – Rowset	BaseUomISO	String	Base unit of measure in ISO code	3	BASE_UOM_ISO	Optional
Input – Rowset	Plant	String	Plant	4	PLANT	Optional
Input – Rowset	OriginGroup	String	Origin group as subdivision of cost element	4	ORIG_GROUP	Optional
Input – Rowset	OriginMaterial	String	Material-related origin	1	ORIG_MAT	Optional
Input – Rowset	CostObject	String	Cost object	12	COST_OBJ	Optional
Input – Rowset	ProfitCenter	String	Profit Center	10	PROFIT_CTR	Optional
Input – Rowset	PartnerProfitCenter	String	Partner profit center	10	PART_PRCTR	Optional
Input – Rowset	WBSElement	String	Work Breakdown Structure element (WBS element)	24	WBS_ELEMENT	Optional
Input – Rowset	NetworkNo	String	Network number for Account Assignment	12	NETWORK	Optional
Input – Rowset	RoutingNo	String	Routing number of operations in the order	10	ROUTING_NO	Optional
Input – Rowset	OrderItemNo	String	Order Item Number	4	ORDER_ITNO	Optional
Input – Rowset	CommitmentItem	String	Commitment item	14	CMMT_ITEM	Optional
Input – Rowset	FundsCenter	String	Funds center	16	FUNDS_CTR	Optional
Input – Rowset	Fund	String	Fund	10	FUND	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	SalesOrderNo	String	Sales order number	10	SALES_ORD	Optional
Input – Rowset	SalesOrderItemNo	String	Item number in Sales Order	6	S_ORD_ITEM	Optional
Input – Rowset	TradingPartnerCompanyId	String	Company ID of trading partner	6	TRADE_ID	Optional
Input – Rowset	ValuationArea	String	Valuation area	4	VAL_AREA	Optional
Input – Rowset	ValuationType	String	Valuation type	10	VAL_TYPE	Optional
Input – Rowset	PreviousDocRefProcedure	String	Previous document: Reference procedure	5	OBJ_TYP_P	Optional
Input – Rowset	PreviousDocObjKey	String	Previous document: Object key (AWREF_REV and AWORG_REV)	20	OBJ_KEY_P	Optional
Input – Rowset	PreviousDocRefLineItem	String	Previous document: Reference line item	10	OBJ_POS_P	Optional
Input – Rowset	ItemCategory	String	Item Category in Purchasing Document	1	ITEM_CAT	Optional
Input – Rowset	DebitCreditInd_GL	String	Indicator: subsequent debit/credit	1	DE_CRE_IND	Optional
Input – Rowset	MaterialType	String	Material type	4	MATL_TYPE	Optional
Input – Rowset	PartnerProfitCenter_InternalBusinessElimination	String	Partner profit center for elimination of internal business	10	P_EL_PRCTR	Optional
Input – Rowset	ConditionType	String	Condition type	4	COND_TYPE	Optional
Input – Rowset	StepNo	String	Step number	3	COND_ST_NO	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	ConditionCounter	String	Condition counter	2	COND_COUNT	Optional
Input – Rowset	FunctionalArea	String	Functional area	4	FUNC_AREA	Optional
Input – Rowset	UnitOfEntryQuantity	String	Quantity in Unit of Entry	13	ENTRY_QNT	Optional
Input – Rowset	EntryUOM	String	Unit of Entry	3	ENTRY_UOM	Optional
Input – Rowset	InternalEntryUOM	String	The value for this parameter passes to SAP if value for EntryUOM is null or empty	3	ENTRY_UOM	Optional
Input – Rowset	EntryUomISO	String	Unit of entry in ISO code	3	ENTRY_UOM_ISO	Optional
Input – Rowset	ActivityType	String	Activity type	6	ACTTYPE	Optional
Input – Rowset	BusinessProcess	String	Business process	12	CO_BUSPROC	Optional
Input – Rowset	ActivityNo	String	Operation/Activity Number	4	ACTIVITY	Optional
Input – Rowset	MaterialNoLong	String	Long Material Number for MATERIAL Field	40	MATERIAL_EXTERNAL	Optional
Input – Rowset	MaterialGUID	String	External GUID for MATERIAL field	32	MATERIAL_GUID	Optional
Input – Rowset	MaterialVersionNo	String	Version number for MATERIAL Field	10	MATERIAL_VERSION	Optional
Input – Rowset	FunctionalAreaLong	String	Functional area	16	FUNC_AREA_LONG	Optional
Input – Rowset	CommitmentItemLong	String	Commitment item	24	CMMT_ITEM_LONG	Optional
Input – Rowset	GrantNo	String	Grant	20	GRANT_NBR	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	Segment	String	Segment for Segmental Reporting	10	SEGMENT	Optional
Input – Rowset	PartnerSegment	String	Partner segment for Segmental Reporting	10	PARTNER_SEGMENT	Optional
Input – Rowset	BudgetPeriod	String	FM: Budget Period	10	BUDGET_PERIOD	Optional
Input – Rowset	PPAExcludeInd	String	PPA Exclude Indicator	1	PPA_EX_IND	Optional
Input – Parameter	TaxDataLineList	Array of rows	This array of parameters is used for non-PO invoices.			Mandatory
Input – Rowset	Match_TAX	Boolean	If the value is set to TRUE, the corresponding row is included for non-PO invoice creation.			Optional
Input – Rowset	AccDocLineItemNo_Tax	Integer	Accounting document Line Item Number	10	ITEMNO_ACC	Mandatory
Input – Rowset	GLAccountNo	Integer	General ledger account	10	GL_ACCOUNT	Mandatory
Input – Rowset	TaxCode	Integer	Tax on sales/purchases code	2	TAX_CODE	Mandatory
Input – Rowset	Transaction Key	Integer	Transaction Key	3	ACCT_KEY	Optional
Input – Rowset	TaxJurisdiction	String	Tax Jurisdiction	15	TAXJURCODE	Mandatory
Input – Rowset	ConditionType	String	Condition Type	4	COND_KEY	Optional
Input – Rowset	TaxRate	String	Tax rate	7	TAX_RATE	Optional
Input – Rowset	TaxDate	String	Date for defining tax rates	8	TAX_DATE	Optional
Input – Rowset	StatisticsCondition	String	Condition used for statistics	1	STAT_CON	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	TaxJurCodeDeep	String	Tax Jurisdiction Code - Jurisdiction for lowest level tax	15	TAXJURCODE_DEEP	Optional
Input – Rowset	TaxJurCodeLevel	String	Tax jurisdiction code level	1	TAXJURCODE_LEVEL	Optional
Input – Rowset	NegetivePostingInd	String	Indicator for negative Posting	1	NEG_POSTNG	Optional
Input – Parameter	CurrencyLineList	Array of rows	This array of parameters is used for non-PO invoices.			Mandatory
Input – Rowset	Match_CURR	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.			Optional
Input – Rowset	AccDocLineItemNo_CURR	Integer	Accounting document line item number	10	ITEMNO_ACC	Mandatory
Input – Rowset	CurrencyType	String	Currency type and valuation view	2	CURR_TYPE	Optional
Input – Rowset	CurrencyKey	String	Currency key	5	CURRENCY	Optional
Input – Rowset	CurrencyISO	String	ISO code currency	3	CURRENCY_ISO	Mandatory
Input – Rowset	DocumentCurrencyAmount	Double	Amount in document currency	23	AMT_DOCCUR	Mandatory
Input – Rowset	ExchangeRate	String	Exchange rate	9	EXCH_RATE	Optional
Input – Rowset	TaxBaseAmount	String	Tax Base Amount in Document Currency	23	AMT_BASE	Mandatory
Input – Rowset	DiscountBase	String	Amount eligible for cash discount in Document Currency	23	DISC_BASE	Optional
Input – Rowset	IndirectQuotedExchangeRate	String	Indirect Quoted Exchange Rate	9	EXCH_RATE_V	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Parameter	POLineList	Array of rows	This array of parameters is used for Non-PO invoices			Optional
Input – Rowset	ReferenceTransaction	String	Reference transaction	5	OBJ_TYPE	Optional
Input – Rowset	ReferenceKey	String	Reference key	20	OBJ_KEY	Optional
Input – Rowset	SourceDocSystem	String	Logical system of source document	10	OBJ_SYS	Optional
Input – Rowset	ReferenceLineItem	String	Reference line item	10	OBJ_POS	Optional
Input – Rowset	ReferenceDocumentId	String	Reference document: Current identification	10	OBJ_ID	Optional
Input – Rowset	DocDate	Date	Document Date	8	DOC_DATE	Optional
Input – Rowset	VendorAccNumber	String	Vendor Account Number	10	VENDOR	Optional
Input – Rowset	TradingPartnerId	String	Company ID of trading partner	6	TRADE_ID	Optional
Input – Rowset	TradingPartnerBusinessArea	String	Trading partner's business area	4	TR_PART_BA	Optional
Input – Rowset	PODate	Date	Order date of schedule line	8	PO_DATE	Optional
Input – Rowset	DeliveryDate	Date	Delivery date	8	DELIV_DATE	Optional
Input – Rowset	PrevDocReferenceProcedure	String	Previous document: Reference procedure	5	OBJ_TYP_P	Optional
Input – Rowset	PrevDocObjKey	String	Previous document: Object key (AWREF_REV and AWORG_REV)	20	OBJ_KEY_P	Optional
Input – Rowset	PrevDocReferenceLineItem	String	Previous document: Reference line item	10	OBJ_POS_P	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	CompanyCode	String	Company Code	4	COMP_CODE	Optional
Input – Rowset	BusinessArea	String	Business Area	4	BUS_AREA	Optional
Input – Rowset	GLAccount	String	G/L Account Number	10	GL_ACCOUNT	Optional
Input – Rowset	OriginGroup	String	Origin group as subdivision of Cost Element	4	ORIG_GROUP	Optional
Input – Rowset	Network	String	Network number for Account Assignment	12	NETWORK	Optional
Input – Rowset	OrderId	String	Order Number	12	ORDERID	Optional
Input – Rowset	WbsElement	String	Work Breakdown Structure element (WBS element)	24	WBS_ELEMENT	Optional
Input – Rowset	AssetNo	String	Main Asset Number	12	ASSET_NO	Optional
Input – Rowset	AssetSubNo	String	Asset Subnumber	4	SUB_NUMBER	Optional
Input – Rowset	CostCenter	String	Cost center	10	COSTCENTER	Optional
Input – Rowset	PaymentTerms	String	Terms of Payment Key	4	PMNTTRMS	Optional
Input – Rowset	DiscountDays1	String	Cash discount days 1	3	DSCT_DAYS1	Optional
Input – Rowset	DiscountDays2	String	Cash discount days 2	3	DSCT_DAYS2	Optional
Input – Rowset	NetTerms	String	Net payment terms period	3	NETTERMS	Optional
Input – Rowset	DiscountPercentage1	String	Cash discount percentage 1	5	DSCT_PCT1	Optional
Input – Rowset	MaterialNo	String	Material number	18	MATERIAL	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	MaterialGroup	String	Material group	9	MATL_GROUP	Optional
Input – Rowset	SegmentText	String	Segment text	50	SEG_TEXT	Optional
Input – Rowset	GoodsReceivedQuantity	String	Quantity of goods received	13	DELIV_QTY	Optional
Input – Rowset	BaseUOM	String	Base unit of measure	3	BASE_UOM	Optional
Input – Rowset	BaseUOM_ISO	String	Base unit of measure in ISO code	3	BASE_UOM_ISO	Optional
Input – Rowset	DeleteIndicator	String	Deletion indicator in Purchasing Document	1	DELETE_IND	Optional
Input – Rowset	CommitmentItem	String	Commitment item	14	CMMT_ITEM	Optional
Input – Rowset	Fund	String	Fund	10	FUND	Optional
Input – Rowset	FundsCenter	String	Funds center	16	FUNDS_CTR	Optional
Input – Rowset	PurchaseQuantity	String	Quantity moved in GR/GI	13	PURCHASE_QTY	Optional
Input – Rowset	InvoiceQuantity	String	Quantity invoiced	13	INVOICE_QTY	Optional
Input – Rowset	MaterialNoLong	String	Long material number for MATERIAL field	40	MATERIAL_EXTERNAL	Optional
Input – Rowset	MaterialGUID	String	External GUID for MATERIAL field	32	MATERIAL_GUID	Optional
Input – Rowset	MaterialVersion	String	Version number for MATERIAL field	10	MATERIAL_VERSION	Optional
Input – Rowset	DiscountPercentage2	String	Cash discount percentage 2	5	DSCT_PCT2	Optional
Input – Rowset	CommitmentItemLong	String	Commitment item	24	CMMT_ITEM_LONG	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	Grant	String	Grant	20	GRANT_NBR	Optional
Input – Rowset	FunctionalAreaLong	String	Functional area	16	FUNC_AREA_LONG	Optional
Input – Rowset	Activity	String	Operation/Activity number	4	ACTIVITY	Optional
Input – Rowset	EarmarkedFundsDocNo	String	Document number for Earmarked Funds	10	FUNDS_RES	Optional
Input – Rowset	EarmarkedFundsDocItem	String	Earmarked Funds: Document Item	3	RES_ITEM	Optional
Input – Rowset	ScheduleLineNo	String	Delivery Schedule line number	4	SCHED_LINE	Optional
Input – Rowset	FundLong	String	Long Fund (Obsolete)	20	FUND_LONG	Optional
Input – Rowset	NoMoreGR	String	"Delivery Completed" indicator	1	NO_MORE_GR	Optional
Input – Rowset	FinalInvoiceIndicator	String	Final invoice indicator	1	FINAL_INV	Optional
Input – Rowset	GoodsReceiptIndicator	String	Goods receipt indicator	1	GR_IND	Optional
Input – Rowset	GoodsReceiptNonValuated	String	Goods receipt, non-valuated	1	GR_NON_VAL	Optional
Input – Rowset	InvoiceReceiptIndicator	String	Invoice receipt indicator	1	IR_IND	Optional
Input – Rowset	BudgetPeriod	String	Budget period	10	BUDGET_PERIOD	Optional
Input – Rowset	POType	String	Order type (Purchasing)	4	PO_TYPE	Optional
Input – Parameter	PurchaseAmountList	Array of rows	This array of parameters is used for Non-PO invoices			Optional
Input – Rowset	ReferenceTransaction	String	Reference Transaction	5	OBJ_TYPE	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	ReferenceKey	String	Reference Key	20	OBJ_KEY	Optional
Input – Rowset	SourceDocSystem	String	Logical system of source document	10	OBJ_SYS	Optional
Input – Rowset	ReferenceLineItem	String	Reference line item	10	OBJ_POS	Optional
Input – Rowset	ReferenceDocumentId	String	Reference document: Current identification	10	OBJ_ID	Optional
Input – Rowset	OrderValueType	String	Value type of purchase order/sales order values	1	ORVALTYPE	Optional
Input – Rowset	CurrencyType	String	Currency type and valuation view	2	CURR_TYPE	Optional
Input – Rowset	CurrencyKey	String	Currency key	5	CURRENCY	Optional
Input – Rowset	CurrencyISO	String	ISO code currency	3	CURRENCY_ISO	Optional
Input – Rowset	OriginalAmount	String	Original amount	23	AMT_ORIGIN	Optional
Input – Rowset	OpenAmount	String	Open amount	23	AMT_OPEN	Optional
Input – Rowset	ReducedAmount	String	Reduced amount	23	AMT_ACTUAL	Optional
Input – Rowset	ExchangeRate	String	Exchange rate	9	EXCH_RATE	Optional
Input – Rowset	IndirectQuotedExchangeRate	String	Indirect quoted exchange rate	9	EXCH_RATE_V	Optional
Input – Rowset	ScheduleLineNo	String	Delivery schedule line number	4	SCHED_LINE	Optional
Input – Parameter	COPACriteriaList	Array of rows	This array of parameters is used for Non-PO invoices.			Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	Match_COPACriteria	Boolean	If the value is set to TRUE, the corresponding row is included for non-PO invoice creation.			Optional
Input – Rowset	AccDocLineItemNo_COPACriteria	Integer	Accounting document line item number	10	ITEMNO_ACC	Mandatory
Input – Rowset	FieldName	String	Field name	30	FIELDNAME	Optional
Input – Rowset	Characteristic	String	Characteristic (concrete use - EDIMAP)	18	CHARACTER	Optional
Input – Rowset	COPAValueFieldLineList	Array of rows	This array of parameters is used for Non-PO invoices.			Optional
Input – Rowset	Match_COPAValue	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.			Optional
Input – Rowset	AccDocLineItemNo_COPAVal	Integer	Accounting document line item number	10	ITEMNO_ACC	Mandatory
Input – Rowset	FieldName	String	Field Name	30	FIELDNAME	Optional
Input – Rowset	CurrencyType	String	Currency type and valuation view	2	CURR_TYPE	Optional
Input – Rowset	CurrencyKey	String	Currency Key	5	CURRENCY	Optional
Input – Rowset	CurrencyISO	String	ISO code currency	3	CURRENCY_ISO	Optional
Input – Rowset	CurrencyValueComponent	String	Value components: Currency field	23	AMT_VALCOM	Optional
Input – Rowset	BaseUom	String	Base unit of measure	3	BASE_UOM	Optional

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name	Requirement
Input – Rowset	BaseUomISO	String	Base unit of measure in ISO code	3	BASE_UOM_ISO	Optional
Input – Rowset	QuantitiesValueField	String	Value fields: Quantities	15	QUA_VALCOM	Optional
Input – Parameter	ExtensionLineList	Array of rows	This array of parameters is used for Non-PO invoices			Optional
Input – Rowset	Match_EXTENSIONIN	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.			Optional
Input – Rowset	Field1	String	Character field with length 250	250	FIELD1	Optional
Input – Rowset	Field2	String	Character field with length 250	250	FIELD2	Optional
Input – Rowset	Field3	String	Character field with length 250	250	FIELD3	Optional
Input – Rowset	Field4	String	Character field with length 250	250	FIELD4	Optional

The following output mapping XML is used to configure the output parameters for CreateNonPOInvoice operation.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- @REP_VERSION_NUMBER@ -->
<c:outputs xmlns:in="http://www.perceptivesoftware.com/pif/imagenow"
xmlns:c="http://www.perceptivesoftware.com/pif/mapping">
    <c:parameter>
        <c:name>InvoiceDocNumber</c:name>
        <in:objectPropertyWriter>
            <in:name></in:name>
            <in:objectId></in:objectId>
            <in:objectType></in:objectType>
            <in:propertyType></in:propertyType>
        </in:objectPropertyWriter>
    </c:parameter>
    <c:parameter>
        <c:name>FiscalYear</c:name>
        <in:objectPropertyWriter>
            <in:name></in:name>
            <in:objectId></in:objectId>
            <in:objectType></in:objectType>
            <in:propertyType></in:propertyType>
        </in:objectPropertyWriter>
    </c:parameter>
       <c:parameter>
            <c:name>CompanyCode</c:name>
            <in:objectPropertyWriter>
                <in:name></in:name>
                <in:objectId></in:objectId>
                <in:objectType></in:objectType>
                <in:propertyType></in:propertyType>
            </in:objectPropertyWriter>
        </c:parameter>
    <c:parameter>
      <c:name>IsDuplicateInvoice</c:name>
      <in:objectPropertyWriter>
        <in:name></in:name>
        <in:objectId></in:objectId>
        <in:objectType></in:objectType>
        <in:propertyType></in:propertyType>
      </in:objectPropertyWriter>
    </c:parameter>
</c:outputs>
```

The table given below lists the output parameters of the XML file.

Parameter type	Parameter name	Data type	Description	Field Length	SAP Parameter name
Output	InvoiceDocNumber	String	Invoice document number	10	
Output	FiscalYear	String	Fiscal year	4	
Output	CompanyCode	String	Company code	4	
Output	IsDuplicateInvoice	Boolean	Duplicate invoice indicator is set to TRUE when soft stop is enabled and a duplicate invoice is detected based on your configuration.	1	

Synchronous operations

About synchronous operations

The synchronous operations available with Perceptive Connector for SAP Financials enable you to synchronously extract the required information from the associated SAP application.

For synchronous calls, ErpId passes as a SOAP parameter. If you do not provide any value for this parameter, the default connection is used.

ErpSyncConnector-<version>.jar: GetCompanyDetails

GetCompanyDetails returns company details based on the company code you specify.

The connector operations use ErpId for the SAP connection. This operation calls the following function module.

- BAPI_COMPANYCODE_GETDETAIL

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetCompanyDetails>
         <ErpId></ErpId>
         <CompanyCode></CompanyCode>
      </syn:GetCompanyDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetCompanyDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CompanyCode	String	Company code	4	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCompanyDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <Company>
            <CompanyCode></CompanyCode>
            <CompanyName></CompanyName>
            <City></City>
            <LanguageCode></LanguageCode>
            <CountryKey></CountryKey>
            <CurrencyCode></CurrencyCode>
            <Vat/>
         </Company>
      </ns1:GetCompanyDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of Company.

Parameter name	Data type	Description	Field length
CompanyCode	String	Company code	4
CompanyName	String	Name of company code or company	25
City	String	City	25
LanguageCode	String	Language key	2
CountryKey	String	Country key	2
CurrencyCode	String	Currency key	3
VAT	String	Tax on sales/purchases code	20

ErpSyncConnector-<version>.jar: GetCompanyList

GetCompanyList returns a list of companies based on search parameters you specify.

Search returns partial matches too. This operation calls the following function module.

- BAPI_COMPANYCODE_GETLIST

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued in the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetCompanyList>
         <ErpId></ErpId>
         <VendorGroup></VendorGroup>
         <CompanyName></CompanyName>
         <CompanyCode></CompanyCode>
      </syn:GetCompanyList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetCompanyList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
VendorGroup	String	Vendor group	For future use	
CompanyName	String	Name of company	25	Optional
CompanyCode	String	Company code	4	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCompanyListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <Company>
            <CompanyCode> </CompanyCode>
            <CompanyName> </CompanyName>
            <VendorGroup> </VendorGroup>
         </Company>
      </ns1:GetCompanyListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of Company.

Parameter name	Data type	Description	Field length
CompanyCode	String	Company code	4
CompanyName	String	Name of company code or company	25
VendorGroup	String	For future use	

ErpSyncConnector-<version>.jar: GetCostCenterDetails

GetCostCenterDetails returns the details of cost centers in a controlling area and company. If you specify a language, the search returns results in that language. The default language is the language that you configured in Configure SAP connectivity.

The connector operations use ErpId for the SAP connection. This operation calls the following function modules.

- BAPI_COSTCENTER_GETDETAIL1
- RFC_READ_TABLE
  - Table TKA02

Ensure that you perform the following step when you request for a response.

- You must provide a value for the controlling area, or the company code, or both. If you keep both fields empty, the search does not return any result.

The following example displays the XML template and parameter list issued in the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:getCostCenterDetails>
         <ErpId></ErpId>
         <ControllingArea></ControllingArea>
         <CompanyCode></CompanyCode>
         <CostCenterId></CostCenterId>
         <ValidOnDate></ValidOnDate>
         <Language></Language>
      </syn:getCostCenterDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:getCostCenterDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use. For sync calls, ErpId passes as a SOAP parameter. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.	10	Optional
ControllingArea	String	Controlling area is an organizational unit from the accounting component in the R/3 System.	4	Optional
CompanyCode	String	Company or business unit code	4	Optional
CostCenterId	String	Cost center ID	10	Mandatory
ValidOnDate	Date	Valid on date	8 for yyyyMMdd 10 for yyyy-MM-dd	Optional
Language	String	Language according to ISO 639	2	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCostCentersResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <CostCenter>
            <costCenter></costCenter>
            <name></name>
            <description></description>
            <controllingArea></controllingArea>
            <details>
               <validFromDate></validFromDate>
               <validToDate></validToDate>
               <personInCharge></personInCharge>
               <personInChargeUserId/>
               <companyCode></companyCode>
               <department></department>
               <businessArea></businessArea>
               <costCenterType></costCenterType>
               <currency></currency>
               <costCenterHierarchyGroup></costCenterHierarchyGroup>
               <profitCenter></profitCenter>
               <address></address>
               <functionalArea></functionalArea>
            </details>
         </CostCenter>
      </ns1:GetCostCentersResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of costCenter.

Parameter name	Data type	Description	Field length
costCenter	String	Cost center	10
name	String	Name of the contact person	
description	String	Description of the cost center	
controllingArea	String	Controlling area is an organizational unit from the accounting component in the R/3 System.	4
details		This is an object name (Refer to the table below for parameters under this object)	

The following table describes the parameters of details under CostCenter.

Parameter name	Data type	Description	Field length
validFromDate	Date	The date from which this cost center is valid	8
validToDate	Date	The date until which this cost center is valid	8
personInCharge	String	Person in charge for the cost center	20
personInChargeUserId	String	User ID of person in charge	12
companyCode	String	Company or business unit code	4
department	String	Department	12
businessArea	String	Business area	4
costCenterType	String	Type of cost center	1
currency	String	Currency	3
costCenterHierarchyGroup	String	Cost center hierarchy group	12
profitCenter	String	Profit center	10

Parameter name	Data type	Description	Field length
address	String	Address	250
functionalArea	String	Functional area	4

ErpSyncConnector-<version>.jar: GetCostCenters

GetCostCenters returns the list of cost centers based on controlling area, company code, or both. You can further filter the search by cost center range, date range, and cost center group.

This operation calls the following function modules.

- BAPI_COSTCENTER_GETLIST1
- RFC_READ_TABLE
    - Table TKA02

The connector operations use ErpId for the SAP connection.

If you provide only CostCenterFrom the search returns only that cost center under the specified business unit. Otherwise, the search returns results for the range of cost centers you provide.

If you specify a date range the search returns all valid results falling within that date range, regardless of the date range length. If you do not provide ValidFromDate, the application refers to the current date by default. Supported date formats for ValidFromDate and ValidToDate are yyyy-MM-dd and yyyyMMdd.

If you set IncludeDetails to TRUE, the search returns the cost center details for the cost centers listed under the controlling area you specify. You can specify the number of results you want from this search in MaxNumberOfHits.

Ensure the following steps when you request for a response.

- You must provide a value for controlling area, company code, or both. If you keep both the fields empty, the search does not return any result.
- Enter a part of cost center ID in CostCenterIdPartial and the search returns matching results.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetCostCenters>
         <ErpId></ErpId>
         <ControllingArea></ControllingArea>
         <CompanyCode></CompanyCode>
         <CostCenterPartial></CostCenterPartial>
         <CostCenterFrom></CostCenterFrom>
         <CostCenterTo></CostCenterTo>
         <ValidFromDate></ValidFromDate>
         <ValidToDate></ValidToDate>
         <CostCenterGroup></CostCenterGroup>
         <IncludeDetails></IncludeDetails>
         <MaxNumberOfHits></MaxNumberOfHits>
      </syn:GetCostCenters>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetCostCenters.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
ControllingArea	String	Controlling area is an organizational unit from the accounting component in the R/3 System.	4	Optional
CompanyCode	String	Company or business unit code	4	Optional
CostCenterPartial	String	Cost center ID for partial search	10	Optional
CostCenterFrom	String	Cost center (from value)	10	Optional
CostCenterTo	String	Cost center (to value)	10	Optional
ValidFromDate	Date	Start of the validity period	8 for yyyyMMdd 10 for yyyy-MM-dd	Optional
ValidToDate	Date	End of the validity period	8 for yyyyMMdd 10 for yyyy-MM-dd	Optional
CostCenterGroup	String	Cost center group	15	Optional
IncludeDetails	Boolean	Set TRUE to get details	4	Optional
MaxNumberOfHits	Integer	Maximum number of search results	4	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCostCentersResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <CostCenter>
            <costCenter></costCenter>
            <name></name>
            <description></description>
            <controllingArea></controllingArea>
            <details>
               <validFromDate></validFromDate>
               <validToDate></validToDate>
               <personInCharge></personInCharge>
               <personInChargeUserId/>
               <companyCode></companyCode>
               <department></department>
               <businessArea></businessArea>
               <costCenterType></costCenterType>
               <currency></currency>
               <costCenterHierarchyGroup></costCenterHierarchyGroup>
               <profitCenter></profitCenter>
               <address></address>
               <functionalArea></functionalArea>
            </details>
         </CostCenter>
      </ns1:GetCostCentersResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of CostCenter.

Parameter name	Data type	Description	Field length
costCenter	String	Cost center	4
name	String	Name of the cost center	20
description	String	Description of the cost center	40
controllingArea	String	Controlling area is an organizational unit from the accounting component in the R/3 System.	4
details		This is an object name (Refer to the table below for parameters under this object)	

The following table describes the parameters of details under CostCenter.

Parameter name	Data type	Description	Field length
validFromDate	Date	The date from which this cost center is valid	8
validToDate	Date	The date to which this cost center is valid	8
personInCharge	String	Person in charge for the cost center	20
personInChargeUserId	String	User ID of the person in charge	12
companyCode	String	Company or business unit code	4
department	String	Department	12

Parameter name	Data type	Description	Field length
businessArea	String	Business area	4
costCenterType	String	Type of cost center	1
currency	String	Currency	3
costCenterHierarchyGroup	String	Cost center hierarchy group	12
profitCenter	String	Profit center	10
address	String	Address	250
functionalArea	String	Functional area	4

ErpSyncConnector-<version>.jar: GetCurrenciesForCountry

GetCurrenciesForCountry returns the list of currencies available for the country key you provide.

This operation calls the following function module.

- RFC_READ_TABLE
    - Table T500W
    - Table TCURT
    - Table T002

The connector operations use ErpId for the SAP connection. If you specify a language, the search returns results in that language. The default language is the language that you configured in Configure SAP connectivity.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetCurrenciesForCountry>
         <ErpId></ErpId>
         <CountryKey></CountryKey>
         <Language></Language>
      </syn:GetCurrenciesForCountry>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetCurrenciesForCountry.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CountryKey	String	Country code in ISO alpha 2 format	3	Mandatory
Language	String	Language for currency list.	2	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCurrenciesForCountryResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <CurrencyISOCode></CurrencyISOCode>
         <Description></Description>
         <DescriptionLanguage></DescriptionLanguage>
      </ns1:GetCurrenciesForCountryResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of ns1:GetCurrenciesForCountryResponse.

Parameter name	Data type	Description	Field length
CurrencyISOCode	String	Currency key	3
Description	String	Description	15
DescriptionLanguage	String	Language for description	2

ErpSyncConnector-<version>.jar: GetCurrencyList

GetCurrencyList returns the list of all currencies available. This operation calls the following function module.

- RFC_READ_TABLE
  - Table TCURT
  - Table T002

The connector operations use ErpId for the SAP connection. You can filter the search based on currency code and language. If you specify a language, the search returns results in that language. The default language is the language that you configured in Configure SAP connectivity.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetCurrencyList>
         <ErpId></ErpId>
         <CurrencyCode></CurrencyCode>
         <Language></Language>
      </syn:GetCurrencyList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetCurrencyList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CurrencyCode	String	Currency key	3	Optional
Language	String	Language for currency list.	2	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetCurrencyListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <Currency>
            <CurrencyISOCode></CurrencyISOCode>
            <Description></Description>
         </Currency>
      </ns1:GetCurrencyListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of Currency.

Parameter name	Data type	Description	Field length
CurrencyISOCode	String	Currency key	3
Description	String	Currency code description	40

ErpSyncConnector-<version>.jar: GetEmailAddresses

This web method returns email addresses based on the specified address number.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table ADR6

The connector operations use ErpId for the SAP connection.

Ensure the following step when you request for a response.

- Get the address number from the GetVendorList response.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetEmailAddresses>
         <ErpId></ErpId>
         <AddressNumber></AddressNumber>
      </syn:GetEmailAddresses>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetEmailAddresses.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
AddressNumber	String	Address number	10	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetEmailAddressesResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <EmailAddresses>
            <eMail></eMail>
         </EmailAddresses>
      </ns1:GetEmailAddressesResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of EmailAddresses.

Parameter name	Data type	Description	Field length
EmailAddresses		This is an object name	
eMail	String	SMTP internet mail address (business address services)	241

ErpSyncConnector-<version>.jar: GetErpServiceList

GetErpServiceList returns the list of available ERP services. This web service does not have any input parameters.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetErpServiceList/>
   </soapenv:Body>
</soapenv:Envelope>
```

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetErpServiceListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <ErpServiceList></ErpServiceList>
         <ErpServiceList></ErpServiceList>
      </ns1:GetErpServiceListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameter of ns1:GetErpServiceListResponse.

Parameter name	Data type	Description
ErpServiceList	String	ERP service

ErpSyncConnector-<version>.jar: GetGLAccounts

GetGLAccounts returns the list of General Ledger (GL) accounts available for the company code you specify. This operation calls the following function module.

- BAPI_GL_ACC_GETLIST

You can further filter the results of this search by name and account number. The search returns partial matches for name and account number. The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetGLAccounts>
         <ErpId></ErpId>
         <CompanyCode></CompanyCode>
         <FilterByName></FilterByName>
         <FilterByAcctNumber></FilterByAcctNumber>
      </syn:GetGLAccounts>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetGLAccounts.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CompanyCode	String	Company code	4	Mandatory
FilterByName	String	GL account short text	20	Optional
FilterByAcctNumber	String	GL account number	10	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetGLAccountsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <GLAccount>
            <CompanyNum></CompanyNum>
            <GLAcct></GLAcct>
            <GLAcctDesc></GLAcctDesc>
         </GLAccount>
      </ns1:GetGLAccountsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of GLAccount.

Parameter name	Data type	Description	Field length
CompanyNum	String	Company code	4
GLAcct	String	GL Account number	10
GLAcctDesc	String	GL account description	20

ErpSyncConnector-<version>.jar: GetPaymentTermList

GetPaymentTermList returns details of a payment term based on specified search parameters.

This operation calls the following function modules.

- MSR20_MD_PAYTERMS_GETLIST
- RFC_READ_TABLE
  - Table T002

The search also returns partial matches. If you specify a language, the search returns results in that language. The default language is the language that you configured in Configure SAP connectivity. The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetPaymentTermList>
         <ErpId></ErpId>
         <PaymentTermsCode></PaymentTermsCode>
         <Language></Language>
      </syn:GetPaymentTermList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetPaymentTermList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
PaymentTermsCode	String	Terms of payment key	4	Optional
Language	String	Language for payment terms	2	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetPaymentTermListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <PaymentTerm>
            <Code></Code>
            <Description></Description>
            <CalcMethod></CalcMethod>
            <Days></Days>
         </PaymentTerm>
      </ns1:GetPaymentTermListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of PaymentTerm.

Parameter name	Data type	Description	Field length
Code	String	Terms of payment key	4
Description	String	Description of payment terms	50
CalcMethod	String	For future use	
Days	Integer	For future use	

ErpSyncConnector-<version>.jar: GetProfitCenterList

GetProfitCenterList returns a list of profit centers based on the specified business unit code.

This operation calls the following function modules.

- BAPI_PROFITCENTER_GETLIST
- RFC_READ_TABLE
  - Table TKA02

The connector operations use ErpId for the SAP connection.

Ensure the following step when you request for a response.

- In the ProfitCenterIdPartial, enter a part of profit center ID and the search returns matching results.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetProfitCenterList>
         <ErpId></ErpId>
         <CompanyCode></CompanyCode>
         <ProfitCenterIdPartial></ProfitCenterIdPartial>
      </syn:GetProfitCenterList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetProfitCenterList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CompanyCode	String	Company code	4	Mandatory

ProfitCenterIdPartial	String	Profit center ID for partial search	10	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetProfitCenterListResponse 
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <ProfitCenter>
            <ProfitCenterId></ProfitCenterId>
            <ControllingArea></ControllingArea>
            <ProfitCenterName></ProfitCenterName>
            <InCharge></InCharge>
            <InChargeUser/>
         </ProfitCenter>
      </ns1:GetProfitCenterListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of ProfitCenter.

Parameter name	Data type	Description	Field length
ProfitCenterId	String	Profit center	10
ControllingArea	String	Controlling area	4
ProfitCenterName	String	General name	20
InCharge	String	Person responsible for profit center	20
InChargeUser	String	User responsible for the profit center	12

ErpSyncConnector-<version>.jar: GetPurchaseOrderDetails

GetPurchaseOrderDetails returns details of a particular purchase order based on the specified search parameters and purchase order number.

This operation calls the following function modules.

- BAPI_PO_GETDETAIL
- BAPI_VENDOR_GETDETAIL
- RFC_READ_TABLE
  - Table T006A

The search also returns partial matches.

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
```

```
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetPurchaseOrderDetails>
         <ErpId></ErpId>
         <PONumber></PONumber>
         <CompanyCode></CompanyCode>
         <VendorGroup></VendorGroup>
          <ShowDeletedLines></ShowDeletedLines>
          <ShowLockedLines></ShowLockedLines>
      </syn:GetPurchaseOrderDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetPurchaseOrderDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
PONumber	String	Purchasing document number	10	Mandatory
CompanyCode	String	If you provide a value, it validates whether the PONumber belongs to the company code	4	Optional
VendorGroup	String	Vendor group	For future use	
ShowDeletedLines	Boolean	If the value is set to TRUE, the Purchase Order detail includes the line items deleted in SAP		Mandatory
ShowLockedLines	Boolean	If the value is set to TRUE, the Purchase Order detail includes the line items locked in SAP		Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">/
   <soap:Body>
      <ns1:GetPurchaseOrderDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <PurchaseOrderDetails>
            <VendorList>
               <VendorNo/>
               <VendorName/>
               <Addr1/>
               <City/>
               <State/>
               <Zip/>
               <CountryKey/>
               <POBox/>
            </VendorList>
            <CodingLines>
               <PONumber> </PONumber>
               <LineNumber> </LineNumber>
               <ItemNumber> </ItemNumber>
               <DeleteInd> </DeleteInd>
               <ItemDescription> </ItemDescription>
               <Quantity> </Quantity>
               <OpenQuantity> </OpenQuantity>
               <OpenIVQuantityPO> </OpenIVQuantityPO>
               <UOM> </UOM>
               <InternalUOM> </InternalUOM>
               <POPriceUOM></POPriceUOM>
               <InternalPOPriceUOM></InternalPOPriceUOM>
               <ConvNum></ConvNum>
               <ConvDen></ConvDen>
               <UnitPrice> </UnitPrice>
               <PriceUnit> </PriceUnit>
               <ExtendedAmount> </ExtendedAmount>
               <POLineTaxable> </POLineTaxable>
               <POLVATJurisdiction> </POLVATJurisdiction>
               <POLVATCode> </POLVATCode>
               <TaxAmount> </TaxAmount>
               <ReferenceDocument></ReferenceDocument>
               <ReferenceDocumentYear></ReferenceDocumentYear>
               <ReferenceDocumentItem></ReferenceDocumentItem>
               <ItemCategory></ItemCategory>
               <DistribIndicator></DistribIndicator>
               <IsBlanketPOItem></IsBlanketPOItem>
               <Material/>
               <TaxCode></TaxCode>
               <TaxJurisdictionCode></TaxJurisdictionCode>
               <WBS/>
               <IsGRBasedInvoice></IsGRBasedInvoice>
               <GRIndicator></GRIndicator>
               <AcctAssignmentCategory></AcctAssignmentCategory>
               <PartialInvoiceIndicator></PartialInvoiceIndicator>
               <GRNonValuated></GRNonValuated>
               <IRIndicator></IRIndicator>
               <DeliveryCompletedIndicator></DeliveryCompletedIndicator>
               <FinalInvoiceIndicator></FinalInvoiceIndicator>
               <TrackingNo></TrackingNo>
               <PurchaseRequisitionNumber></PurchaseRequisitionNumber>
               <PurchaseRequisitionerName></PurchaseRequisitionerName>
            </CodingLines>
            <AccountAssignmentLines>
               <ItemNumber></ItemNumber>
               <SerialNo></SerialNo>
               <DistributionPercentage></DistributionPercentage>
               <ProjectDefinition/>
               <ScheduleLineNo></ScheduleLineNo>
               <AssetNo/>
               <SubNumber/>
               <GoodsRecipient/>
               <UnloadPoint/>
               <ControllingArea></ControllingArea>
               <ToCostCenter/>
               <ToOrder/>
               <ToProject/>
               <GLAccount></GLAccount>
               <CostCenter></CostCenter>
               <SDDoc/>
               <SalesDocItem></SalesDocItem>
               <OrderNo/>
               <CommitmentItem/>
               <FundsCenter/>
               <FunctionalArea/>
               <ProfitCenter></ProfitCenter>
               <Fund/>
               <FundsRes/>
               <ResItem></ResItem>
               <BusArea></BusArea>
               <PartnerAccountNo/>
               <CostObject/>
               <Network/>
               <RoutingNo></RoutingNo>
               <Activity/>
               <RealEstateKey/>
               <Counter></Counter>
               <RecoveryIndicator/>
               <GrantNbr/>
               <CommitmentItemLong/>
               <FuncAreaLong/>
               <Quantity></Quantity>
               <ProfitSegmNumber></ProfitSegmNumber>
               <BudgetPeriod/>
               <WbsElement/>
               <UOM></UOM>
               <POPriceQuantity></POPriceQuantity>
               <POPriceUOM></POPriceUOM>
               <UOMConvFactor></UOMConvFactor>
               <TaxCode></TaxCode>
               <TaxJurisdictionCode></TaxJurisdictionCode>
               <IsBlanketPOItem></IsBlanketPOItem>
            </AccountAssignmentLines>
            <POAmount> </POAmount>
            <POCurrency/>
            <MoreResultsCo></MoreResultsCo>
            <PONumber/>
            <POType></POType>
            <POCategory></POCategory>
            <IsBlanketPO></IsBlanketPO>
            <AcctDataPopulationRequired></AcctDataPopulationRequired>
            <CreatedBy></CreatedBy>
         </PurchaseOrderDetails>
      </ns1:GetPurchaseOrderDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameter of PurchaseOrderDetails.

Parameter name	Data type	Description	Field length
VendorList		This is an object name (Refer to the table below for parameters under this object).	

Parameter name	Data type	Description	Field length
CodingLines		This is an object name (Refer to the table under CodingLines for parameters under this object).	
POAmount	Double	Purchase order amount	
POCurrency	String	Currency key	5
MoreResultsCo	Boolean	For future use	
PONumber	String	Purchasing document number	10
POType	String	Document type (DOC_TYPE) of the PO in SAP	4
POCategory	String	Document category (DOC_CAT) of the PO in SAP	1
IsBlanketPO	Boolean	Indicator for blanket PO	
AcctDataPopulationRequired	Boolean	Indicates whether passing of accounting data to SAP is required while parking or posting invoices	
CreatedBy	String	Creator of the PO	12

The following table describes the parameters of VendorList under PurchaseOrderDetails.

Parameter name	Data type	Description	Field length
VendorNo	String	Account number of vendor or creditor	10
VendorName	String	Vendor name	35
Addr1	String	House number and street	35
City	String	City	35
State	String	Region (state, province, county)	3
Zip	String	Postal code	10
CountryKey	String	Country key	3
POBox	String	PO Box number	10

The following list describes the parameters of CodingLines under PurchaseOrderDetails.

Parameter name	Data type	Description	Field length
PONumber	String	Purchasing document number	10
LineNumber	Integer	Sequential number	
ItemNumber	String	Item number of purchasing document	5
ItemDescription	String	Short text	40
DeleteInd	String	Possible values for this parameter are X, L, S, or blank. X or L denotes that the Line Item of the PO is deleted, S denotes that the Line Item of the PO is locked, and if no value is specified, the Line Item is valid.	1
Quantity	Double	Purchase order quantity	
OpenQuantity	Double	Undelivered quantity of the particular item of a Purchase Order	
OpenIVQuantityPO	Double	Quantity of the particular item of a Purchase Order that is not invoiced	
UOM	String	Purchase order unit of measure	3
InternalUOM	String	Internal unit of measure for SAP	3
POPriceUOM	String	Order Price Unit (Purchasing)	3
InternalPOPriceUOM	String	Internal Order Price Unit (Purchasing)	3
ConvNum	Integer	Numerator for conversion of Order Price Unit into Order Unit	5
ConvDen	Integer	Denominator for conversion of Order Price Unit into Order Unit	5
UnitPrice	Double	Net price in purchasing document (in document currency)	
PriceUnit	Double	Price unit	
ExtendedAmount	Double	Gross order value in PO currency	
POLineTaxable	Boolean	For future use	

Parameter name	Data type	Description	Field length
POLVATJurisdiction	String	Tax jurisdiction	15
POLVATCode	String	Tax on sales/purchases code	2
TaxAmount	Double	Tax amount	
ReferenceDocument	String	Document number of a reference document	10
ReferenceDocumentYear	String	Fiscal year of current period	4
ReferenceDocumentItem	String	Item of a reference document	4
ItemCategory	String	Category of the line item (ITEM_CAT) in SAP	1
DistribIndicator		Distribution indicator for multiple account assignment	
IsBlanketPOItem	Boolean	Indicates a blanket PO item	
Material	String	Material number of the material in purchase order	18
TaxCode	String	Tax code	2
TaxJurisdictionCode	String	Tax jurisdiction code	15
ProfitCenter	String	Profit center, cost center, WBS, and GL Account together referred as account assignment of the particular line item in PO.	10
CostCenter	String		10
GLAccount	String		10
WBS	String		24
IsGRBasedInvoice	Boolean	GR based invoice indicator (GR_BASEDIV) in SAP	
GRIndicator	Boolean	This parameter refers to the Goods Receipt Indicator. The value can be either TRUE or FALSE.	
TrackingNo:	String	Requirement tracking number	10
PurchaseRequisitionNumber	String	Purchase requisition number	10
PurchaseRequisitionName	String	Name of requester	12

The following table contains a list of the parameters of AccountAssignmentLines under PurchaseOrderDetails.

Parameter name	Data type	Description	Field length
ItemNumber	String	Item number of purchasing document	5
SerialNo	String	Sequential number of account assignment	2
DistributionPercentage	String	Distribution percentage in case of multiple account assignment	3
ProjectDefinition	String	Project definition	24
ScheduleLineNo	String	Delivery schedule line number	4
AssetNo	String	Main asset number	12
SubNumber	String	Asset subnumber	4
GoodsRecipient	String	Goods recipient/Ship to party	12
UnloadPoint	String	Unloading point	25
ControllingArea	String	Controlling area	4
ToCostCenter	String	Posting to cost center	1
ToOrder	String	Post To order	1
ToProject	String	Post to project	1
GLAccount	String	G/L account number	10
CostCenter	String	Cost center	10
SDDoc	String	Sales and distribution document number	10
SalesDocItem	String	Sales document item	6
OrderNo	String	Order number	12
CommitmentItem	String	Commitment item	14
FundsCenter	String	Funds center	16
FunctionalArea	String	Functional area	4
ProfitCenter	String	Profit center	10

Parameter name	Data type	Description	Field length
Fund	String	Fund	10
FundsRes	String	Document number for earmarked funds	10
ResItem	String	Earmarked funds: Document item	3
BusArea	String	Business area	4
PartnerAccountNo	String	Partner account number	10
CostObject	String	Cost object	12
Network	String	Network number for account assignment	12
RoutingNo	String	Routing number of operations in the order	10
Activity	String	Operation/Activity number	4
RealEstateKey	String	Internal key for real estate object	8
Counter	String	Internal counter	8
RecoveryIndicator	String	Recovery indicator	2
GrantNbr	String	Grant	20
CommitmentItemLong	String	Commitment item	14
FuncAreaLong	String	Functional area	16
Quantity	Double	Quantity	13
ProfitSegmNumber	String	Profitability Segment Number (CO-PA)	10
BudgetPeriod	String	FM: Budget period	10
WbsElement	String	Work Breakdown Structure Element (WBS Element)	24
UOM	String	Purchase order unit of measure	3
POPriceQuantity	Double	Quantity in purchase order for price unit	13, 3
POPriceUOM	String	Order Price Unit (Purchasing)	3
UOMConvFactor	Double	Conversion factor of Order Price Unit into Order Unit	

Parameter name	Data type	Description	Field length
TaxCode	String	Tax code	2
TaxJurisdictionCode	String	Tax jurisdiction code	15
IsBlanketPOItem	Boolean	Indicates a blanket PO item	

ErpSyncConnector-<version>.jar: GetPurchaseOrderList

GetPurchaseOrderList returns a purchase order list based on the specified search parameters.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table EKKO

The search also returns partial matches. The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetPurchaseOrderList>
         <ErpId></ErpId>
         <CompanyCode></CompanyCode>
         <VendorNo></VendorNo>
      </syn:GetPurchaseOrderList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetPurchaseOrderList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CompanyCode	String	Company code	4	Optional
VendorNo	String	Vendor account number	10	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetPurchaseOrderListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <PurchaseOrder>
            <CompanyCode></CompanyCode>
            <PODate> </PODate>
            <PODateMatched> </PODateMatched>
            <PONumber> </PONumber>
            <POAmount> </POAmount>
            <VendorNo> </VendorNo>
            <POCurrency> </POCurrency>
            <RemitTo/>
         </PurchaseOrder>
      </ns1:GetPurchaseOrderListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of PurchaseOrder.

Parameter name	Data type	Description	Field length
CompanyCode	String	company code	4
PODate	Date	Purchasing document date	8
PODateMatched	Boolean	For future use	
PONumber	String	Purchasing document number	10
POAmount	Double	For future use	
POCurrency	String	Currency key	5
VendorNo	String	Vendor account number	10
RemitTo	String	For future use	

ErpSyncConnector-<version>.jar: GetPurchaseRequisitionDetails

GetPurchaseRequisitionDetails returns the purchase requisition details based on the requisition number.

This operation calls the following function modules.

- BAPI_PR_GETDETAIL
- RFC_READ_TABLE
  - Table T006A

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetPurchaseRequisitionDetails>
```

```
         <ErpId></ErpId>
         <RequisitionNumber></RequisitionNumber>
      </syn:GetPurchaseRequisitionDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetPurchaseRequisitionDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
RequisitionNumber	String	Requisition number	10	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetPurchaseRequisitionDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <PurchaseRequisitionDetails>
            <PRNumber></PRNumber>
            <PRType></PRType>
            <PRLines>
               <PRNumber></PRNumber>
               <ItemNumber></ItemNumber>
               <ReleaseStatus></ReleaseStatus>
               <CreatedBy></CreatedBy>
               <Requisitioner></Requisitioner>
               <Material></Material>
               <Quantity></Quantity>
               <UOM></UOM>
               <InternalUOM> </InternalUOM>
               <ItemCategory></ItemCategory>
               <MRPController></MRPController>
               <MRPArea></MRPArea>
               <PONumber></PONumber>
               <POItem></POItem>
            </PRLines>
         </PurchaseRequisitionDetails>
      </ns1:GetPurchaseRequisitionDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of PurchaseRequisitionDetails.

Parameter name	Data type	Description	Field length

PRNumber	String	Purchase requisition number	10
PRType	String	Requisition type	4
PRLines		This is an object name (Refer to the table under PRLines for parameters under this object)	

The following table describes the parameters of PRLines under PurchaseRequisitionDetails.

Parameter name	Data type	Description	Field length
PRNumber	String	Purchase requisition number	10
ItemNumber	String	The number to which the item belongs in a particular purchase requisition	5
ReleaseStatus	String	Status of item released in SAP	8
CreatedBy	String	Purchase requisition item creator	12
Requisitioner	String	Requester of a particular item in purchase requisition	12
Material	String	Material code in SAP	18
Quantity	String	Quantity specified in purchase requisition	13
UOM	String	Unit of measure	3
InternalUOM	String	Internal unit of measure for SAP	3
ItemCategory	String	Category of an item in SAP	1
MRPController	String	MRP controller	3
MRPArea	String	MRP area	10
PONumber	String	Associated PO number of the purchase requisition item, if the purchase requisition is assigned to a PO.	10
POItem	String	The item number of this item in the PO doc, if this purchase requisition item is already assigned to a PO.	5

ErpSyncConnector-<version>.jar: GetTaxCodes

GetTaxCodes returns the list of tax codes based on the specified country key.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table T005
  - Table T007S
  - Table T002

The connector operations use ErpId for the SAP connection. If you specify a language, the search returns results in that language. The default language is the language that you configured using the ERP SAP Config dialog box.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetTaxCodes>
         <ErpId></ErpId>
         <CountryKey></CountryKey>
         <Language></Language>
      </syn:GetTaxCodes>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetTaxCodes.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CountryKey	String	Country key	3	Mandatory
Language	String	Language key	1	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetTaxCodesResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <TaxCode>
            <Code></Code>
            <Description></Description>
         </TaxCode>
      </ns1:GetTaxCodesResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of TaxCode.

Parameter name	Data type	Description	Field length
Code	String	Tax code	2
Description	String	Description of tax code	50

ErpSyncConnector-<version>.jar: GetVendorDetails

GetVendorDetails returns the details of a vendor based on the specified search parameters.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table LFA1

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetVendorDetails>
         <ErpId></ErpId>
         <VendorNo></VendorNo>
      </syn:GetVendorDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetVendorDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
VendorNo	String	Account number of vendor	10	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetVendorDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <Vendor>
            <VendorNo></VendorNo>
            <VendorName></VendorName>
            <Addr1></Addr1>
            <City></City>
            <State></State>
            <Zip></Zip>
            <CountryKey></CountryKey>
            <POBox></POBox>
            <Telephone><Telephone/>
            <FaxNumber><FaxNumber/>
            <Details>
              <BankAccountNo></BankAccountNo>
              <BankAccountName> </BankAccountName>
              <BankName></BankName>
              <BankCountry></BankCountry>
              <IBAN></IBAN>
              <BankKey></BankKey>
              <BankControlKey></BankControlKey>
              <PartnerBankType></PartnerBankType>
              <SwiftCode></SwiftCode>
              <emailAddressList>
                 <eMail></eMail>
              </emailAddressList>
            </Details>
         </Vendor>
      </ns1:GetVendorDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of Vendor.

Parameter name	Data type	Description	Field length
VendorNo	String	Vendor account number	10
VendorName	String	Vendor name	35
Addr1	String	Street address	35
City	String	City	35
State	String	Regional code	3
Zip	String	Postal code	10
CountryKey	String	Country key	3
POBox	String	PO Box number	10
Telephone	String	Telephone	16
FaxNumber	String	Fax number	31
Details		This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameters of Details under Vendor.

Parameter name	Data type	Description	Field length
BankAccountNo	String	Bank account number	18
BankAccountName	String	Bank account name	35
BankName	String	Bank name	60
BankCountry	String	Bank country key	3
IBAN	String	International Bank Account Number	34
BankKey	String	Bank keys	15
BankControlKey	String	Bank control key	2
PartnerBankType	String	Partner bank type	4
SwiftCode	String	Swift code	11
emailAddressList		This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameter of emailAddressList under Details.

Parameter name	Data type	Description	Field length
eMail	String	SMTP internet mail address (business address services)	241

ErpSyncConnector-<version>.jar: GetVendorList

GetVendorList returns a vendor list based on the specified search parameters.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table LFA1
  - Table LFB1

The search also returns partial matches. The connector operations use Erpld for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetVendorList>
         <ErpId></ErpId>
         <CompanyCode></CompanyCode>
         <VendorGroup></VendorGroup>
         <VendorName></VendorName>
         <VendorNo></VendorNo>
      </syn:GetVendorList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetVendorList.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
CompanyCode	String	Company code	4	Mandatory
VendorGroup	String	Vendor group	For future use	
VendorName	String	Vendor name	35	Optional
VendorNo	String	Vendor account number	10	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetVendorListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <Vendor>
            <VendorNo></VendorNo>
            <VendorName> </VendorName>
            <Addr1> </Addr1>
            <City> </City>
            <State/>
            <Zip/>
            <CountryKey> </CountryKey>
            <AddressNumber> </AddressNumber>
            <POBox> </POBox>
         </Vendor>
      </ns1:GetVendorListResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of Vendor.

Parameter name	Data type	Description	Field length
VendorNo	String	Vendor account number	10
VendorName	String	Vendor name	35
Addr1	String	Street address	35
City	String	City	35
State	String	Regional code	3
Zip	String	Postal code	10
CountryKey	String	Country key	3
AddressNumber	String	Address number	10
POBox	String	PO Box number	10

ErpSyncConnector-<version>.jar: GetVendorRemitToList

GetVendorRemitToList returns the list of remit vendors based on the specified search parameters.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table WYT3

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetVendorRemitToList>
         <ErpId></ErpId>
         <VendorNo></VendorNo>
      </syn:GetVendorRemitToDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetVendorRemitToList.

Parameter name	Data type	Description	Field length	Requirement

ErpId	String	SAP connection that the connector operations use.	10	Optional
VendorNo	String	Vendor account number	10	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetVendorRemitToDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <VendorRemitToList>
            <RemitToVendorNo></RemitToVendorNo>
            <PurchasingOrg></PurchasingOrg>
            <Plant/>
            <VendorDetails>
               <VendorNo></VendorNo>
               <VendorName></VendorName>
               <Addr1></Addr1>
               <City></City>
               <State></State>
               <Zip></Zip>
               <CountryKey></CountryKey>
               <POBox></POBox>
               <Details>
                  <BankAccountNo></BankAccountNo>
                  <BankAccountName> </BankAccountName>
                  <BankName></BankName>
                  <SwiftCode/>
                  <Telephone/>
                  <FaxNumber/>
                  <emailAddressList>
                     <eMail></eMail>
                  </emailAddressList>
               </Details>
            </VendorDetails>
         </VendorRemitToList>
      </ns1:GetVendorRemitToDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of VendorRemitToList.

Parameter name	Data type	Description	Field length
RemitToVendorNo	String	Remit vendor account number	10
PurchasingOrg	String	Purchasing organization	4
Plant	String	An organizational unit serving to sub-divide an enterprise according to production, procurement, maintenance, and materials planning aspects. It is a place where either materials are produced or goods and services provided.	4

Parameter name	Data type	Description	Field length
VendorDetails		This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameters of VendorDetails under VendorRemitToList.

Parameter name	Data type	Description	Field length
VendorNo	String	Vendor account number	10
VendorName	String	Vendor name	35
Addr1	String	Street address	35
City	String	City	35
State	String	Regional code	3
Zip	String	Postal code	10
CountryKey	String	Country key	3
POBox	String	PO Box number	10
Details		This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameters of Details under VendorDetails.

Parameter name	Data type	Description	Field length
BankAccountNo	String	Bank account number	18
Bank AccountName	String	Bank account name	35
BankName	String	Bank name	60
SwiftCode	String	Swift code	11
Telephone	String	Telephone	16
FaxNumber	String	Fax number	31
emailAddressList		This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameter of emailAddressList under Details.

Parameter name	Data type	Description	Field length
eMail	String	SMTP internet mail address (business address services)	241

ErpSyncConnector-<version>.jar: GetWithholdingTaxDetails

GetWithholdingTaxDetails returns the withholding tax details based on the vendor number and company code.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table LFBW

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetWithholdingTaxDetails>
         <!--Optional:-->
         <ErpId></ErpId>
         <!--Optional:-->
         <VendorNo></VendorNo>
         <!--Optional:-->
         <CompanyCode></CompanyCode>
      </syn:GetWithholdingTaxDetails>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetWithholdingTaxDetails.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
VendorNo	String	Vendor account number	10	Optional
CompanyCode	String	Company code	4	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetWithholdingTaxDetailsResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <WithholdingTaxDetail>
            <VendorNo></VendorNo>
            <CompanyCode></CompanyCode>
            <WithholdTaxType></WithholdTaxType>
            <WithholdTaxCode></WithholdTaxCode>
         </WithholdingTaxDetail>
      </ns1:GetWithholdingTaxDetailsResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of WithholdingTaxDetail.

Parameter name	Data type	Description	Field length
VendorNo	String	Vendor account number	10
CompanyCode	String	Company Code	4
WithholdTaxType	String	Indicator for withholding tax type	2
WithholdTaxCode	String	Withholding tax code	2

ErpSyncConnector-<version>.jar: GetVendorCompanyMaster

GetVendorCompanyMaster returns the payment method and payment term of the vendor based on the specified search parameters.

This operation calls the following function module.

- RFC_READ_TABLE
  - Table T001
  - Table T042Z
  - Table LFB1

The connector operations use ErpId for the SAP connection.

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:GetVendorCompanyMaster>
         <!--Optional:-->
         <ErpId></ErpId>
         <Vendor></Vendor>
         <CompanyCode></CompanyCode>
      </syn:GetVendorCompanyMaster>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameters of syn:GetVendorCompanyMaster.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional
Vendor	String	Vendor number	10	Mandatory
CompanyCode	String	Company code	4	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:GetVendorCompanyMasterResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <VendorCompanyMaster>
            <PaymentTerm>ZB01</PaymentTerm>
            <paymentMethod>
               <Code>T</Code>
               <Name>Transferencia Bancaria</Name>
            </paymentMethod>
         </VendorCompanyMaster>
      </ns1:GetVendorCompanyMasterResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of VendorCompanyMaster.

Parameter name	Data type	Description	Field length
PaymentTerm	String	Payment term	4
paymentMethod	Array	This is an object name (Refer to the table below for parameters under this object).	

The following table describes the parameters of paymentMethod under VendorCompanyMaster.

Parameter name	Data type	Description	Field length
Code	String	Terms of payment key	1
Name	String	Name of the vendor	30

ErpSyncConnector-<version>.jar: CreateInvoice and ParkInvoice

The CreateInvoice and ParkInvoice sync operations help create and park, respectively, all types of invoices, including Purchase Order (PO) based invoices, Non-PO based invoices, and Credit Memos. These synchronous operations do not require a mapping XML.

The CreateInvoice operation calls the following function module.

- BAPI_INCOMINGINVOICE_CREATE

The ParkInvoice operation calls the following function module.

- BAPI_INCOMINGINVOICE _PARK

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:CreateInvoice>
         <!--Optional:-->
         <ErpId></ErpId>
         <Invoice>
            <!--Optional:-->
            <EnableDuplicateInvoiceCheck></EnableDuplicateInvoiceCheck>
            <!--Optional:-->
            <InvoiceInd></InvoiceInd>
            <!--Optional:-->
            <DocType></DocType>
            <!--Optional:-->
            <DocDate></DocDate>
            <!--Optional:-->
            <PostingDate></PostingDate>
            <!--Optional:-->
            <RefDocNo></RefDocNo>
            <!--Optional:-->
            <CompanyCode></CompanyCode>
            <!--Optional:-->
            <DiffInvParty></DiffInvParty>
            <!--Optional:-->
            <Currency></Currency>
            <!--Optional:-->
            <CurrencyISO></CurrencyISO>
            <!--Optional:-->
            <DirectExchRate></DirectExchRate>
            <!--Optional:-->
            <IndirectExchRate></IndirectExchRate>
            <!--Optional:-->
            <GrossAmount></GrossAmount>
            <!--Optional:-->
            <TaxAutoCalcInd></TaxAutoCalcInd>
            <!--Optional:-->
            <PaymentTerm></PaymentTerm>
            <!--Optional:-->
            <BaselineDate></BaselineDate>
            <!--Optional:-->
            <CashDiscDays1></CashDiscDays1>
            <!--Optional:-->
            <CashDiscDays2></CashDiscDays2>
            <!--Optional:-->
            <NetPaymentTermPeriod></NetPaymentTermPeriod>
            <!--Optional:-->
            <CashDiscPerc1></CashDiscPerc1>
            <!--Optional:-->
            <CashDiscPerc2></CashDiscPerc2>
            <!--Optional:-->
            <IVCategory></IVCategory>
            <!--Optional:-->
            <HeaderText></HeaderText>
            <!--Optional:-->
            <PaymentBlock></PaymentBlock>
            <!--Optional:-->
            <DeliveryCosts></DeliveryCosts>
            <!--Optional:-->
            <DeliveryCostTaxCode></DeliveryCostTaxCode>
            <!--Optional:-->
            <DeliveryJurisdictionCode></DeliveryJurisdictionCode>
            <!--Optional:-->
            <ExtUserName></ExtUserName>
            <!--Optional:-->
            <PaymentMethod></PaymentMethod>
            <!--Optional:-->
            <PaymentMethodSupplement></PaymentMethodSupplement>
            <!--Optional:-->
            <InvDocNo></InvDocNo>
            <!--Optional:-->
            <SCBankIndicator></SCBankIndicator>
            <!--Optional:-->
            <SupplyingCountry></SupplyingCountry>
            <!--Optional:-->
            <ServiceIndicator></ServiceIndicator>
            <!--Optional:-->
            <RefDocNoLong></RefDocNoLong>
            <!--Optional:-->
            <CashDiscAmt></CashDiscAmt>
            <!--Optional:-->
            <POSubscriberNo></POSubscriberNo>
            <!--Optional:-->
            <POCheckDigit></POCheckDigit>
            <!--Optional:-->
            <POReferenceNo></POReferenceNo>
            <!--Optional:-->
            <Payee></Payee>
            <!--Optional:-->
            <PartnerBankType></PartnerBankType>
            <!--Optional:-->
            <HouseBankId></HouseBankId>
            <!--Optional:-->
            <AssignmentNo></AssignmentNo>
            <!--Optional:-->
            <PaymentReference></PaymentReference>
            <!--Optional:-->
            <InvRefNo></InvRefNo>
            <!--Optional:-->
            <InvYear></InvYear>
            <!--Optional:-->
            <InvRecDate></InvRecDate>
            <!--Optional:-->
            <PlanningLevel></PlanningLevel>
            <!--Optional:-->
            <PlanningDate></PlanningDate>
            <!--Optional:-->
            <FixedPaymentTerm></FixedPaymentTerm>
            <!--Optional:-->
            <BusinessLocation></BusinessLocation>
            <!--Optional:-->
            <LotNumber></LotNumber>
            <!--Optional:-->
            <ItemText></ItemText>
            <!--Optional:-->
            <FiscalType></FiscalType>
            <!--Optional:-->
            <EUTriangularDeal></EUTriangularDeal>
            <!--Optional:-->
            <ReportingCountry></ReportingCountry>
            <!--Optional:-->
            <VATRegNo></VATRegNo>
            <!--Optional:-->
            <BusinessPlace></BusinessPlace>
            <!--Optional:-->
            <TaxExchRate></TaxExchRate>
            <!--Optional:-->
            <GoodsAffectedIndicator></GoodsAffectedIndicator>
            <!--Optional:-->
            <RetDueDate></RetDueDate>
            <!--Optional:-->
            <PostingLogicForDelivery></PostingLogicForDelivery>
            <!--Optional:-->
            <PostingLogicForReturn></PostingLogicForReturn>
            <!--Zero or more repetitions:-->
            <InvoiceLineItemList>
               <!--Optional:-->
               <Match_INV></Match_INV>
               <!--Optional:-->
               <InvoiceDocItem_INV></InvoiceDocItem_INV>
               <!--Optional:-->
               <PONumber></PONumber>
               <!--Optional:-->
               <POItem></POItem>
               <!--Optional:-->
               <RefDocNo_INV></RefDocNo_INV>
               <!--Optional:-->
               <RefDocYear></RefDocYear>
               <!--Optional:-->
               <RefDocItem></RefDocItem>
               <!--Optional:-->
               <DebitCreditInd_INV></DebitCreditInd_INV>
               <!--Optional:-->
               <TaxCode></TaxCode>
               <!--Optional:-->
               <TaxJurCode></TaxJurCode>
               <!--Optional:-->
               <ItemAmount></ItemAmount>
               <!--Optional:-->
               <Quantity></Quantity>
               <!--Optional:-->
               <UOM></UOM>
               <!--Optional:-->
               <InternalUOM></InternalUOM>
               <!--Optional:-->
               <UnitISO></UnitISO>
               <!--Optional:-->
               <POPriceQuantity></POPriceQuantity>
               <!--Optional:-->
               <POPriceUOM></POPriceUOM>
               <!--Optional:-->
               <InternalPOPriceUOM></InternalPOPriceUOM>
               <!--Optional:-->
               <POPriceUomISO></POPriceUomISO>
               <!--Optional:-->
               <ConditionType></ConditionType>
               <!--Optional:-->
               <ConditionStepNo></ConditionStepNo>
               <!--Optional:-->
               <ConditionCount></ConditionCount>
               <!--Optional:-->
               <SheetNo></SheetNo>
               <!--Optional:-->
               <ItemText></ItemText>
               <!--Optional:-->
               <FinalInvoiceIndicator></FinalInvoiceIndicator>
               <!--Optional:-->
               <LineNo></LineNo>
               <!--Optional:-->
               <GRIRClearingInd></GRIRClearingInd>
               <!--Optional:-->
               <VendorFreight></VendorFreight>
               <!--Optional:-->
               <CashDiscInd></CashDiscInd>
               <!--Optional:-->
               <RetentionAmount></RetentionAmount>
               <!--Optional:-->
               <RetentionPercentage></RetentionPercentage>
               <!--Optional:-->
               <RetentionDueDate></RetentionDueDate>
               <!--Optional:-->
               <IrrelevantRetention></IrrelevantRetention>
               <!--Optional:-->
               <ValuationType></ValuationType>
            </InvoiceLineItemList>
            <!--Zero or more repetitions:-->
            <InvoiceGLLineItemList>
               <!--Optional:-->
               <Match_GL>false</Match_GL>
               <!--Optional:-->
               <InvoiceDocItem_GL></InvoiceDocItem_GL>
               <!--Optional:-->
               <GLAccount></GLAccount>
               <!--Optional:-->
               <ItemAmount></ItemAmount>
               <!--Optional:-->
               <DebitCreditInd_GL></DebitCreditInd_GL>
               <!--Optional:-->
               <NegPosting></NegPosting>
               <!--Optional:-->
               <CompanyCode></CompanyCode>
               <!--Optional:-->
               <TaxCode></TaxCode>
```

```
              <!--Optional:-->
              <TaxJurCode></TaxJurCode>
              <!--Optional:-->
              <ItemText></ItemText>
              <!--Optional:-->
              <CostCenter></CostCenter>
              <!--Optional:-->
              <SDDoc></SDDoc>
              <!--Optional:-->
              <SalesDocItem></SalesDocItem>
              <!--Optional:-->
              <Orderid></Orderid>
              <!--Optional:-->
              <RefDate></RefDate>
              <!--Optional:-->
              <CommitmentItem></CommitmentItem>
              <!--Optional:-->
              <FundsCenter></FundsCenter>
              <!--Optional:-->
              <FunctionalArea></FunctionalArea>
              <!--Optional:-->
              <ProfitCenter></ProfitCenter>
              <!--Optional:-->
              <Fund></Fund>
              <!--Optional:-->
              <BusArea></BusArea>
              <!--Optional:-->
              <TradingPartBussArea></TradingPartBussArea>
              <!--Optional:-->
              <Costobject></Costobject>
              <!--Optional:-->
              <Network></Network>
              <!--Optional:-->
              <Activity></Activity>
              <!--Optional:-->
              <WbsElement></WbsElement>
              <!--Optional:-->
              <ActivityType></ActivityType>
              <!--Optional:-->
              <RealEstateKey></RealEstateKey>
              <!--Optional:-->
              <PersonnelNumber></PersonnelNumber>
              <!--Optional:-->
              <CoBusProc></CoBusProc>
              <!--Optional:-->
              <GrantNbr></GrantNbr>
              <!--Optional:-->
              <CommitmentItemLong></CommitmentItemLong>
              <!--Optional:-->
              <FuncAreaLong></FuncAreaLong>
              <!--Optional:-->
              <Quantity></Quantity>
              <!--Optional:-->
              <BaseUOM></BaseUOM>
              <!--Optional:-->
              <AllocNumber></AllocNumber>
              <!--Optional:-->
              <CashDiscIndicator></CashDiscIndicator>
              <!--Optional:-->
              <TaxBaseAmount></TaxBaseAmount>
```

```
               <!--Optional:-->
               <ProfitSegmNumber></ProfitSegmNumber>
               <!--Optional:-->
               <Plant></Plant>
               <!--Optional:-->
               <BudgetPeriod></BudgetPeriod>
            </InvoiceGLLineItemList>
            <InvoiceTaxLineItemList>
               <!--Optional:-->
               <Match_TAX></Match_TAX>
               <!--Optional:-->
               <InvoiceDocItem_TAX></InvoiceDocItem_TAX>
               <!--Optional:-->
               <TaxCode></TaxCode>
               <!--Optional:-->
               <TaxAmount></TaxAmount>
               <!--Optional:-->
               <TaxBaseAmount></TaxBaseAmount>
               <!--Optional:-->
               <TaxJurisdiction></TaxJurisdiction>
               <!--Optional:-->
               <TaxjurcodeDeep></TaxjurcodeDeep>
               <!--Optional:-->
               <CondType></CondType>
            </InvoiceTaxLineItemList>
            <!--Zero or more repetitions:-->
            <InvoiceWithholdingTaxLineItemList>
               <!--Optional:-->
               <Match_WITHHOLDING_TAX></Match_WITHHOLDING_TAX>
               <!--Optional:-->
               <InvoiceDocItem_WITHHOLDING_TAX></InvoiceDocItem_WITHHOLDING_TAX>
               <!--Optional:-->
               <SplitKey></SplitKey>
               <!--Optional:-->
               <WiTaxType></WiTaxType>
               <!--Optional:-->
               <WiTaxCode></WiTaxCode>
               <!--Optional:-->
               <WiTaxBase></WiTaxBase>
               <!--Optional:-->
               <WiTaxAmt></WiTaxAmt>
               <!--Optional:-->
               <WiTaxWithheldAmt></WiTaxWithheldAmt>
            </InvoiceWithholdingTaxLineItemList>
         </Invoice>
      </syn:CreateInvoice>
   </soapenv:Body>
</soapenv:Envelope>
```

The following table describes the parameter of syn:CreateInvoice.

Parameter name	Data type	Description	Field length	Requirement
ErpId	String	SAP connection that the connector operations use.	10	Optional

The following table describes the parameters of Invoice under syn:CreateInvoice.

Parameter name	Data type	Description	Field length	Requirement
EnableDuplicateInvoiceCheck	Boolean	If the value is set to TRUE, the duplicate invoice check is enabled at invoice level. To disable duplicate invoice check, set the value to FALSE. Default value is FALSE.		Optional
InvoiceInd	String	This is a flag to indicate the type of invoice. Possible invoice types X = Regular Invoice “ ” = Credit memo	1	Mandatory
DocType	String	Document Type. Typically RE, but you can customize a doc type. It should consist of two characters only.	2	Optional
DocDate	Date	Date when the document is created.	8	Mandatory
PostingDate	Date	Date when the document is posted in SAP. In the absence of a posting date, the application takes the system date on which the invoice is posted.	8	Optional
RefDocNo	String	This is the Invoice Number. The string has a limit of 16 characters, so invoice numbers with long formats may be recorded elsewhere.	16	Optional
CompanyCode	String	Company code/ Business unit ID	4	Mandatory
DiffInvParty	String	Vendor ID	10	Optional
Currency	String	Local currency for invoice generation	5	Mandatory
CurrencyISO	String	Currency code in ISO 4217 format	3	Optional
IndirectExchRate	Double	Used within Europe for exchange rates with the Euro (EUR)	9,5	Optional
GrossAmount	Double	Total amount in invoice	23,4	Mandatory

Parameter name	Data type	Description	Field length	Requirement
TaxAutoCalcInd	Boolean	Indicator for auto calculate tax applicability		Optional
PaymentTerm	String	Payment term applied for the invoice	4	Optional
BaselineDate	Date	Date from which the system can apply the periods of the terms of payment to calculate cash discount amounts and the due date of an invoice	8	Optional
CashDiscDays1	Double	Cash discount days 1	3	Optional
CashDiscDays2	Double	Cash discount days 2	3	Optional
NetPaymentTermPeriod	Double	Net payment term period	3	Optional
CashDiscPerc1	Double	Cash discount percentage 1	5,3	Optional
CashDiscPerc2	Double	Cash discount percentage 2	5,3	Optional
IVCategory	String	Source of a logistics invoice verification document	1	Optional
HeaderText	String	Document header text	25	Optional
PaymentBlock	String	Payment block key	1	Optional
DeliveryCosts	Double	Unplanned delivery costs	23,4	Optional
DeliveryCostTaxCode	String	Tax code for delivery costs	2	Optional
DeliveryJurisdictionCode	String	Jurisdiction code for delivery costs	15	Optional
ExtUserName	String	Entered by external system user	12	Optional
PaymentMethod	String	Payment method	1	Optional
PaymentMethodSupplement	String	Payment method supplement	2	Optional
InvDocNo	String	Document number of an invoice document	10	ParkInvoice – Mandatory CreateInvoice – Optional
SCBankIndicator	String	State central bank indicator	3	Optional

Parameter name	Data type	Description	Field length	Requirement
SupplyingCountry	String	Supplying country	3	Optional
ServiceIndicator	String	Service indicator for foreign payment	1	Optional
RefDocNoLong	String	Reference document number	35	Optional
CashDiscAmt	Double	Cash discount amount in document currency	23,4	Optional
POSubscriberNo	String	ISR Subscriber number	11	Optional
POCheckDigit	String	ISR check digit	2	Optional
POReferenceNo	String	ISR reference number	27	Optional
Payee	String	Payee/payer	10	Optional
PartnerBankType	String	Partner bank type	4	Optional
HouseBankId	String	Short key for a house bank	5	Optional
AssignmentNo	String	Assignment number	18	Optional
PaymentReference	String	Payment reference	30	Optional
InvRefNo	String	Document number for invoice reference	10	Optional
InvYear	String	Fiscal year of the relevant invoice (for credit memo)	4	Optional
InvRecDate	Date	Invoice receipt date	8	Optional
PlanningLevel	String	Planning level	2	Optional
PlanningDate	Date	Planning date	8	Optional
FixedPaymentTerm	String	Fixed payment terms	1	Optional
BusinessLocation	String	Business area	4	Optional
LotNumber	String	Lot Number for Documents	10	Optional
ItemText	String	Item text	50	Optional
FiscalType	String	Fiscal type	2	Optional

Parameter name	Data type	Description	Field length	Requirement
EUTriangularDeal	String	Indicates a triangular deal within the EU Delivery of goods as part of a triangular deal must be identified when entering such information in the line item and must be listed separately in the EU sales list.	1	Optional
ReportingCountry	String	Reporting country for delivery of goods within the EU	3	Optional
VATRegNo	String	For every EU company that carries out transactions with other EU member states, you should specify the VAT registration number on invoices and purchase orders.	20	Optional
BusinessPlace	String	Business place	4	Optional
TaxExchRate	Double	Exchange rate for taxes	9	Optional
GoodsAffectedIndicator	String	Indicator for capital goods affected	1	Optional
RetDueDate	Date	Due date for retention	8	Optional
PostingLogicForDelivery	String	Invoice / credit memo posting logic for delivery items	1	Optional
PostingLogicForReturn	String	Invoice / credit memo posting logic for return items	1	Optional
InvoiceLineItemList	Array of rows	This array of parameters is used for PO invoices only. (Refer to the table below for parameters under this array)		
InvoiceGLLineItemList	Array of rows	This array of parameters is used for PO invoices only. (Refer to the table below for parameters under this array)		
InvoiceTaxLineItemList	Array of rows	This array of parameters is used for PO and non-PO invoices. To enable InvoiceTaxLineItemList, set the value of TaxAutoCalcInd parameter to FALSE or leave it as blank. If the value of TaxAutoCalcInd parameter is set to TRUE, InvoiceTaxLineItemList is ignored. (Refer to the table below for parameters under this array)		

The following table describes the parameters of InvoiceLineItemList under Invoice.

Parameter name	Data type	Description	Field length	Requirement
Match_INV	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1	Optional
InvoiceDocItem_INV	Integer	Document item in invoice document. If this value is set to 0, the application auto-populates an incremental number while posting to SAP.	6	Optional
PONumber	String	Purchase order number	10	Mandatory
POItem	Long	Item number of purchasing document	5	Mandatory
RefDocNo_INV	String	Document number of a reference document. This is the Invoice Number. The string has a limit of 16 characters, so invoice numbers with long formats may be recorded elsewhere.	10	Optional
RefDocYear	Integer	Fiscal year of current period	4	ParkInvoice – Mandatory CreateInvoice – Optional
RefDocItem	Integer	Item of a reference document	4	ParkInvoice – Mandatory CreateInvoice – Optional
DebitCreditInd_INV	String	Indicator for subsequent debit / credit	1	Optional
TaxCode	String	Tax code	2	ParkInvoice – Optional CreateInvoice – Mandatory
TaxJurCode	String	Tax jurisdiction	15	ParkInvoice – Optional CreateInvoice – Mandatory
ItemAmount	Double	Amount in document currency	23,4	Mandatory
Quantity	Long	Quantity of the line item in a purchase order	13,3	Mandatory

Parameter name	Data type	Description	Field length	Requirement
UOM	String	Unit of measure for purchase order	3	Mandatory
InternalUOM	String	Internal unit of measure for purchase order in SAP	10	Optional
UnitISO	String	ISO codes for standard measurement units are used for data exchange via Electronic Data Interchange (EDI) with external systems. To implement communication with external systems, you must assign ISO codes to SAP internal UOM.	3	Optional
POPriceQuantity	Double	Quantity in purchase order for price unit	13,3	Mandatory
POPriceUOM	String	Order price unit for purchasing	3	Optional
InternalPOPriceUOM	String	Internal Order Price Unit (Purchasing)	3	
POPriceUomISO	String	Purchase order price unit in ISO code	3	Optional
ConditionType	String	Condition type	4	ParkInvoice – Mandatory CreateInvoice – Optional
ConditionStepNo	Integer	Step number	3	Optional
ConditionCount	Integer	Condition counter	2	ParkInvoice – Mandatory CreateInvoice – Optional
SheetNo	Integer	Entry sheet number	10	Optional
ItemText	String	Item text	50	Optional
FinalInvoiceIndicator	String	Indicates that the last invoice from a vendor has been received for an item and no further invoice is expected	1	Optional
LineNo	Long	Line number	10	ParkInvoice – Mandatory CreateInvoice – Optional
GRIRClearingInd	String	Clearing indicator for GR/IR posting for external services	1	Optional

Parameter name	Data type	Description	Field length	Requirement
VendorFreight	String	Account number of vendor or creditor	10	Optional
CashDiscInd	String	Indicator for line item not liable to cash discount	1	Optional
RetentionAmount	Double	Retention amount in document currency	13,2	Optional
RetentionPercentage	Double	Retention in percent	5,2	Optional
RetentionDueDate	Date	Due date for retention	8	Optional
IrrelevantRetention	String	Item not relevant to retention	1	Optional
ValuationType	String	Valuation types are present for each valuation category	10	Optional
ItemCategory	String	Invoice line item category. This value is obtained from GetPurchaseOrderDetails lookup.		

The following table describes the parameters of InvoiceGLLineItemList under Invoice.

Parameter name	Data type	Description	Field length	Requirement
Match_GL	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1	Optional
InvoiceDocItem_GL	Numeric	Document item in invoice document	6	Mandatory
GLAccount	String	General ledger account	10	Mandatory
ItemAmount	Double	Amount in document currency	23,4	Mandatory
DebitCreditInd_GL	String	Indicates debit or credit	1	Mandatory
NegPosting	String	Negative posting indicator	1	Optional
CompanyCode	String	Company code/ business unit ID	4	Mandatory
TaxCode	String	Tax on sales/purchases code	2	Mandatory
TaxJurCode	String	Tax jurisdiction	15	Mandatory
ItemText	String	Item text	50	Optional

Parameter name	Data type	Description	Field length	Requirement
CostCenter	String	Cost center	10	Optional
SDDoc	String	Sales and distribution document number	10	Optional
SalesDocItem	Numeric	Sales document item	6	Optional
Orderid	String	Order number	12	Optional
RefDate	Date	Reference date for settlement	8	Optional
CommitmentItem	String	Commitment item	14	Optional
FundsCenter	String	Funds center	16	Optional
FunctionalArea	String	Functional area	4	Optional
ProfitCenter	String	Profit center	10	Optional
Fund	String	Funds balance provided for a specific purpose by a sponsor and managed separately. Funds are used to represent funds from secondary sources.	10	Optional
BusArea	String	Business area	4	Optional
TradingPartBussArea	String	Business area of trading partner	4	Optional
Costobject	String	Cost object	12	Optional
Network	String	Network number for account assignment	12	Optional
Activity	String	Operation/ activity number	4	Optional
WbsElement	String	Work breakdown structure element (WBS element)	24	Optional
ActivityType	String	Activity type	6	Optional
RealEstateKey	String	Internal key for real estate object	8	Optional
PersonnelNumber	Numeric	Personnel number	8	Optional
CoBusProc	String	Business process	12	Optional
GrantNbr	String	Grant	20	Optional
CommitmentItemLong	String	Commitment item	24	Optional

Parameter name	Data type	Description	Field length	Requirement
FuncAreaLong	String	Functional area	16	Optional
Quantity	Double	Quantity of the line item in a purchase order	13,3	Mandatory
BaseUOM	String	Base unit of measure	3	Mandatory
AllocNumber	String	Assignment number	18	Optional
CashDiscIndicator	String	Cash discount indicator	1	Optional
TaxBaseAmount	Double	Tax base amount in document currency	23,4	Optional
ProfitSegmNumber	Numeric	Profitability segment number (CO-PA)	10	Optional
Plant	String	An organizational unit serving to subdivide an enterprise according to production, procurement, maintenance, and materials planning aspects. It is a place where either materials are produced or goods and services provided.	4	Optional
BudgetPeriod	String	Financial management - period of available budget authority	10	Optional

The following table describes the parameters of InvoiceTaxLineItemList under Invoice.

Parameter name	Data type	Description	Field length	Requirement
Match_TAX	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.	1	Optional
InvoiceDocItem_TAX	Numeric	Document item in invoice document	6	Optional
TaxCode	String	Tax on sales or purchases code	2	Mandatory
TaxAmount	Double	Tax amount in document currency	23	Mandatory
TaxBaseAmount	Double	Tax base amount in document currency	23	Mandatory
TaxJurisdiction	String	Tax jurisdiction	15	Optional
TaxJurCodeDeep	String	Tax Jurisdiction Code - Jurisdiction for Lowest Level Tax	15	

CondType	String	Condition type	4	

The following table describes the parameters of InvoiceWithholdingTaxLineItemList under Invoice.

Parameter name	Data type	Description	Field length	Requirement
Match_WITHHOLDING_TAX	Boolean	If the value is set to TRUE, the corresponding row is included in invoice creation and parking.		
InvoiceDocItem_WITHHOLDING_TAX	Numeric	Document item for withholding tax in invoice document		
SplitKey	String	Grouping key for splitting vendor line item	6	Optional
WiTaxType	String	Indicator for withholding tax type	2	Mandatory
WiTaxCode	String	Withholding tax code	2	Mandatory
WiTaxBase	Double	Withholding tax base amount	23	Mandatory
WiTaxAmt	Double	Withholding tax amount (in Document Currency)	23	Optional
WiTaxWithheldAmt	Double	Withheld withholding tax amount (in document currency)	23	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:CreateInvoiceResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <InvoiceUploadResponse>
            <InvoiceDocNumber></InvoiceDocNumber>
            <FiscalYear></FiscalYear>
            <IsDuplicateInvoice></IsDuplicateInvoice>
         </InvoiceUploadResponse>
      </ns1:CreateInvoiceResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of InvoiceUploadResponse.

Parameter name	Data type	Description	Field length	Requirement
InvoiceDocNumber	String	Invoice document number	10	

FiscalYear	String	Fiscal year	4	
IsDuplicateInvoice	Boolean	Duplicate invoice indicator		

ErpSyncConnector-<version>.jar: PostInvoice

PostInvoice synchronous operation helps to post the parked invoices in SAP.

This operation does not require a mapping XML. This operation calls the following function module.

- BAPI_INCOMINGINVOICE_POST

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:PostInvoice>
         <!--Optional:-->
         <ErpId></ErpId>
         <InvoiceDocNumber></InvoiceDocNumber>
         <FiscalYear></FiscalYear>
      </syn:PostInvoice>
   </soapenv:Body>
</soapenv:Envelope>
```

The table given below lists the input parameters in the XML file.

Parameter name	Data type	Description	Field Length	Requirement
ErpId		SAP connection that the connector operations use.	10	Optional
InvoiceDocNumber	String	Invoice document number	10	Mandatory
FiscalYear	Integer	Fiscal year	4	Optional

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:PostInvoiceResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <InvoiceUploadResponse>
            <InvoiceDocNumber></InvoiceDocNumber>
            <FiscalYear></FiscalYear>
         </InvoiceUploadResponse>
      </ns1:PostInvoiceResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of InvoiceUploadResponse.

Parameter name	Data type	Description	Field length	Requirement
InvoiceDocNumber	String	Invoice document number	10	
FiscalYear	String	Fiscal year	4	

ErpSyncConnector-<version>.jar: CancelInvoice

CancelInvoice synchronous operation helps to cancel the invoices in SAP. This operation does not require a mapping XML.

This operation calls the following function module.

- BAPI_INCOMINGINVOICE_CANCEL

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:CancelInvoice>
         <!--Optional:-->
         <ErpId></ErpId>
         <InvoiceDocNumber></InvoiceDocNumber>
         <FiscalYear></FiscalYear>
         <ReasonForCancellation></ReasonForCancellation>
      </syn:CancelInvoice>
   </soapenv:Body>
</soapenv:Envelope>
```

The table given below lists the input parameters in the XML file.

Parameter name	Data type	Description	Field Length	Requirement
ErpId		SAP connection that the connector operations use.	10	Optional
InvoiceDocNumber	String	Invoice document number	10	Mandatory
FiscalYear	Integer	Fiscal year	4	Mandatory
ReasonForCancel	String	Reason for cancelling the invoice	2	Mandatory

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
```

```
      <ns1:CancelInvoiceResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <InvoiceCancelResponse>
            <InvoiceCancellationDocNumber/>
            <InvoiceCancellationFiscalYear></InvoiceCancellationFiscalYear>
            <MessageInfoTable>
               <MessageCode></MessageCode>
               <Message></Message>
            </MessageInfoTable>
         </InvoiceCancelResponse>
      </ns1:CancelInvoiceResponse>
   </soap:Body>
</soap:Envelope>
```

The following table describes the parameters of InvoiceCancelResponse.

Parameter name	Data type	Description	Field Length	Requirement
InvoiceCancellation DocNumber	Integer	Document number of the cancelled invoice	10	
InvoiceCancellation FiscalYear	Integer	Fiscal year of the cancelled invoice	4	
MessageInfoTable	Array	Message information table. This array of parameters appear if any message is returned from SAP (Refer to the table below for parameters under this array)		

The following table describes the parameters of MessageInfoTable under InvoiceCancelResponse.

Parameter name	Data type	Description	Field Length	Requirement
MessageCode	String	Message code mostly from SAP		
Message	String	Message associated with the message code		

Add or remove parameters from Input mapping XML of asynchronous operations

Configuration files to add or remove parameters from input mapping XML of asynchronous operations

The input mapping XML of certain asynchronous operations display only the parameters that are frequently used. To avoid displaying a long list of parameters in the input mapping XML, many parameters are made configurable by including them in a separate configuration file. This topic describes how you can add or remove parameters from the input mapping XML using the configuration file associated with an asynchronous operation.

CreateInvoice and ParkInvoice asynchronous operations

BAPI_INCOMINGINVOICE_CREATE_OR_PARK.XML is the configuration file associated with the CreateInvoice and ParkInvoice asynchronous operations. After you unzip the PerceptiveConnectorForSAPFinancials-<version>.zip file during connector installation, copy the configuration file from the path [*drive:*]\{*directory where the ZIP file for Perceptive Connector for SAP Financials is extracted*}\SampleCustomSAPOperationMapping\BAPIMapping\AsynchOperationMapping to the path [*drive:*]\{*directory where Perceptive Connect Runtime is extracted*}\Perceptive Connect Runtime\BAPIMapping\AsynchOperationMapping.

The following XML template enables you to add or remove parameters from the input mapping XML of the CreateInvoice and ParkInvoice asynchronous operations.

```
<?xml version="1.0" encoding="utf-8"?>
<CustomizableMappingFields>
  <ADDRESSDATA mapped="false">
    <param name="ACC_1_TIME" mapped="false"></param>
    <param name="NAME" mapped="false"></param>
    <param name="NAME_2" mapped="false"></param>
    <param name="NAME_3" mapped="false"></param>
    <param name="NAME_4" mapped="false"></param>
    <param name="POSTL_CODE" mapped="false"></param>
    <param name="CITY" mapped="false"></param>
    <param name="COUNTRY" mapped="false"></param>
    <param name="STREET" mapped="false"></param>
    <param name="PO_BOX" mapped="false"></param>
    <param name="POBX_PCD" mapped="false"></param>
    <param name="POBK_CURAC" mapped="false"></param>
    <param name="BANK_ACCT" mapped="false"></param>
    <param name="BANK_NO" mapped="false"></param>
    <param name="BANK_CTRY" mapped="false"></param>
    <param name="TAX_NO_1" mapped="false"></param>
    <param name="TAX_NO_2" mapped="false"></param>
    <param name="SLS_PUR_TX" mapped="false"></param>
    <param name="EQUAL_TAX" mapped="false"></param>
    <param name="REGION" mapped="false"></param>
    <param name="CTRL_KEY" mapped="false"></param>
    <param name="INSTR_KEY" mapped="false"></param>
    <param name="DME_IND" mapped="false"></param>
    <param name="LANGU" mapped="false"></param>
    <param name="ENCR_PAYM_RECIP" mapped="false"></param>
    <param name="TAX_CATEGORY" mapped="false"></param>
    <param name="TAX_NO_TY" mapped="false"></param>
    <param name="SOLE_PROP" mapped="false"></param>
    <param name="TAX_NO_3" mapped="false"></param>
    <param name="TAX_NO_4" mapped="false"></param>
    <param name="BANK_REF" mapped="false"></param>
    <param name="REP_NAME" mapped="false"></param>
    <param name="BUS_TYPE" mapped="false"></param>
    <param name="IND_TYPE" mapped="false"></param>
    <param name="FORM_OF_ADDR" mapped="false"></param>
    <param name="VAT_REG_NO" mapped="false"></param>
  </ADDRESSDATA>
  <ACCOUNTINGDATA mapped="false">
    <param name="INVOICE_DOC_ITEM" mapped="false"></param>
    <param name="XUNPL" mapped="false"></param>
    <param name="SERIAL_NO" mapped="false"></param>
    <param name="TAX_CODE" mapped="false"></param>
    <param name="TAXJURCODE" mapped="false"></param>
    <param name="ITEM_AMOUNT" mapped="false"></param>
    <param name="QUANTITY" mapped="false"></param>
    <param name="PO_UNIT" mapped="false"></param>
    <param name="InternalUOM" mapped="false"></param>
    <param name="PO_UNIT_ISO" mapped="false"></param>
    <param name="PO_PR_QNT" mapped="false"></param>
    <param name="PO_PR_UOM" mapped="false"></param>
    <param name="InternalPOPriceUOM" mapped="false"></param>
    <param name="PO_PR_UOM_ISO" mapped="false"></param>
    <param name="GL_ACCOUNT" mapped="false"></param>
    <param name="COSTCENTER" mapped="false"></param>
    <param name="SD_DOC" mapped="false"></param>
    <param name="SDOC_ITEM" mapped="false"></param>
    <param name="ASSET_NO" mapped="false"></param>
    <param name="SUB_NUMBER" mapped="false"></param>
    <param name="ORDERID" mapped="false"></param>
    <param name="REF_DATE" mapped="false"></param>
    <param name="CMMT_ITEM" mapped="false"></param>
    <param name="FUNDS_CTR" mapped="false"></param>
    <param name="FUNC_AREA" mapped="false"></param>
    <param name="FUND" mapped="false"></param>
    <param name="BUS_AREA" mapped="false"></param>
    <param name="RL_EST_KEY" mapped="false"></param>
    <param name="CO_AREA" mapped="false"></param>
    <param name="COSTOBJECT" mapped="false"></param>
    <param name="NETWORK" mapped="false"></param>
    <param name="ACTIVITY" mapped="false"></param>
    <param name="PROFIT_SEGM_NO" mapped="false"></param>
    <param name="PROFIT_CTR" mapped="false"></param>
    <param name="WBS_ELEM" mapped="false"></param>
    <param name="GRANT_NBR" mapped="false"></param>
    <param name="CMMT_ITEM_LONG" mapped="false"></param>
    <param name="FUNC_AREA_LONG" mapped="false"></param>
    <param name="BUDGET_PERIOD" mapped="false"></param>
    <param name="DEL_CREATE_DATE" mapped="false"></param>
  </ACCOUNTINGDATA>
  <MATERIALDATA mapped="false">
    <param name="INVOICE_DOC_ITEM" mapped="false"></param>
    <param name="MATERIAL" mapped="false"></param>
    <param name="VAL_AREA" mapped="false"></param>
    <param name="VALUATION_TYPE" mapped="false"></param>
    <param name="DB_CR_IND" mapped="false"></param>
    <param name="ITEM_AMOUNT" mapped="false"></param>
    <param name="QUANTITY" mapped="false"></param>
    <param name="BASE_UOM" mapped="false"></param>
    <param name="BASE_UOM_ISO" mapped="false"></param>
    <param name="TAX_CODE" mapped="false"></param>
    <param name="TAXJURCODE" mapped="false"></param>
    <param name="MATERIAL_EXTERNAL" mapped="false"></param>
    <param name="MATERIAL_GUID" mapped="false"></param>
    <param name="MATERIAL_VERSION" mapped="false"></param>
  </MATERIALDATA>
  <WITHTAXDATA mapped="false">
    <param name="SPLIT_KEY" mapped="false"></param>
    <param name="WI_TAX_TYPE" mapped="false"></param>
    <param name="WI_TAX_CODE" mapped="false"></param>
    <param name="WI_TAX_BASE" mapped="false"></param>
    <param name="WI_TAX_AMT" mapped="false"></param>
    <param name="WI_TAX_WITHHELD_AMT" mapped="false"></param>
  </WITHTAXDATA>
  <VENDORITEMSPLITDATA mapped="false">
    <param name="SPLIT_KEY" mapped="false"></param>
    <param name="SPLIT_AMOUNT" mapped="false"></param>
    <param name="PYMT_METH" mapped="false"></param>
    <param name="TAX_CODE" mapped="false"></param>
    <param name="PMTMTHSUPL" mapped="false"></param>
    <param name="PMNTTRMS" mapped="false"></param>
  </VENDORITEMSPLITDATA>
  <EXTENSIONIN mapped="false">
    <param name="STRUCTURE" mapped="false"></param>
    <param name="VALUEPART1" mapped="false"></param>
    <param name="VALUEPART2" mapped="false"></param>
    <param name="VALUEPART3" mapped="false"></param>
    <param name="VALUEPART4" mapped="false"></param>
  </EXTENSIONIN>
  <TM_ITEMDATA mapped="false">
    <param name="INVOICE_DOC_ITEM" mapped="false"></param>
    <param name="TOR_NUMBER" mapped="false"></param>
    <param name="TOR_ITEM" mapped="false"></param>
    <param name="LOG_SYSTEM" mapped="false"></param>
    <param name="PO_NUMBER" mapped="false"></param>
    <param name="PO_ITEM" mapped="false"></param>
    <param name="AMT_DOCCUR" mapped="false"></param>
    <param name="TAX_CODE" mapped="false"></param>
    <param name="TAXJURCODE" mapped="false"></param>
    <param name="DE_CRE_IND" mapped="false"></param>
    <param name="ITEM_TEXT" mapped="false"></param>
    <param name="FINAL_INV" mapped="false"></param>
    <param name="CSHDIS_IND" mapped="false"></param>
  </TM_ITEMDATA>
</CustomizableMappingFields>
```

In the configuration file, set the `mapped` attribute of the element and its required parameters to TRUE to include them in the input mapping XML. By default, this value is set to FALSE.

CreateNonPOInvoice asynchronous operation

BAPI_ACC_INVOICE_RECEIPT_POST.XML is the configuration file associated with the CreateNonPOInvoice asynchronous operation. You can locate this file in the path [*drive:*]\{*directory where the ZIP file for Perceptive Connector for SAP Financials is extracted*}\SampleCustomSAPOperationMapping\BAPIMapping\AsynchOperationMapping.

The following XML template enables you to add or remove parameters from the input mapping XML of the CreateNonPOInvoice asynchronous operation.

```
<?xml version="1.0" encoding="utf-8"?>
<CustomizableMappingFields>
       <CUSTOMERCPD mapped="false">
              <param name="NAME" mapped="true"></param>
              <param name="NAME_2" mapped="true"></param>
              <param name="NAME_3" mapped="true"></param>
              <param name="NAME_4" mapped="true"></param>
              <param name="POSTL_CODE" mapped="true"></param>
              <param name="CITY" mapped="true"></param>
              <param name="COUNTRY" mapped="true"></param>
              <param name="COUNTRY_ISO" mapped="true"></param>
              <param name="STREET" mapped="true"></param>
              <param name="PO_BOX" mapped="true"></param>
              <param name="POBX_PCD" mapped="true"></param>
              <param name="POBK_CURAC" mapped="true"></param>
              <param name="BANK_ACCT" mapped="true"></param>
              <param name="BANK_NO" mapped="true"></param>
              <param name="BANK_CTRY" mapped="true"></param>
              <param name="BANK_CTRY_ISO" mapped="true"></param>
              <param name="TAX_NO_1" mapped="true"></param>
              <param name="TAX_NO_2" mapped="true"></param>
              <param name="TAX" mapped="true"></param>
              <param name="EQUAL_TAX" mapped="true"></param>
              <param name="REGION" mapped="true"></param>
              <param name="CTRL_KEY" mapped="true"></param>
              <param name="INSTR_KEY" mapped="true"></param>
              <param name="DME_IND" mapped="true"></param>
              <param name="LANGU_ISO" mapped="true"></param>
              <param name="IBAN" mapped="true"></param>
       </CUSTOMERCPD>
       <PURCHASEORDER mapped="false">
              <param name="OBJ_TYPE" mapped="false"></param>
              <param name="OBJ_KEY" mapped="false"></param>
              <param name="OBJ_SYS" mapped="false"></param>
              <param name="OBJ_POS" mapped="false"></param>
              <param name="OBJ_ID" mapped="false"></param>
              <param name="DOC_DATE" mapped="false"></param>
              <param name="VENDOR" mapped="false"></param>
              <param name="TRADE_ID" mapped="false"></param>
              <param name="TR_PART_BA" mapped="false"></param>
              <param name="PO_DATE" mapped="false"></param>
              <param name="DELIV_DATE" mapped="false"></param>
              <param name="OBJ_TYP_P" mapped="false"></param>
              <param name="OBJ_KEY_P" mapped="false"></param>
              <param name="OBJ_POS_P" mapped="false"></param>
              <param name="COMP_CODE" mapped="false"></param>
              <param name="BUS_AREA" mapped="false"></param>
              <param name="GL_ACCOUNT" mapped="false"></param>
              <param name="ORIG_GROUP" mapped="false"></param>
              <param name="NETWORK" mapped="false"></param>
              <param name="ORDERID" mapped="false"></param>
              <param name="WBS_ELEMENT" mapped="false"></param>
              <param name="ASSET_NO" mapped="false"></param>
              <param name="SUB_NUMBER" mapped="false"></param>
              <param name="COSTCENTER" mapped="false"></param>
              <param name="PMNTTRMS" mapped="false"></param>
              <param name="DSCT_DAYS1" mapped="false"></param>
              <param name="DSCT_DAYS2" mapped="false"></param>
              <param name="NETTERMS" mapped="false"></param>
              <param name="DSCT_PCT1" mapped="false"></param>
              <param name="MATERIAL" mapped="false"></param>
              <param name="MATL_GROUP" mapped="false"></param>
              <param name="SEG_TEXT" mapped="false"></param>
              <param name="DELIV_QTY" mapped="false"></param>
              <param name="BASE_UOM" mapped="false"></param>
              <param name="BASE_UOM_ISO" mapped="false"></param>
              <param name="DELETE_IND" mapped="false"></param>
              <param name="CMMT_ITEM" mapped="false"></param>
              <param name="FUND" mapped="false"></param>
              <param name="FUNDS_CTR" mapped="false"></param>
              <param name="PURCHASE_QTY" mapped="false"></param>
              <param name="INVOICE_QTY" mapped="false"></param>
              <param name="MATERIAL_EXTERNAL" mapped="false"></param>
              <param name="MATERIAL_GUID" mapped="false"></param>
              <param name="MATERIAL_VERSION" mapped="false"></param>
              <param name="DSCT_PCT2" mapped="false"></param>
              <param name="CMMT_ITEM_LONG" mapped="false"></param>
              <param name="GRANT_NBR" mapped="false"></param>
              <param name="FUNC_AREA_LONG" mapped="false"></param>
              <param name="ACTIVITY" mapped="false"></param>
              <param name="FUNDS_RES" mapped="false"></param>
              <param name="RES_ITEM" mapped="false"></param>
              <param name="SCHED_LINE" mapped="false"></param>
              <param name="FUND_LONG" mapped="false"></param>
              <param name="NO_MORE_GR" mapped="false"></param>
              <param name="FINAL_INV" mapped="false"></param>
              <param name="GR_IND" mapped="false"></param>
              <param name="GR_NON_VAL" mapped="false"></param>
              <param name="IR_IND" mapped="false"></param>
              <param name="BUDGET_PERIOD" mapped="false"></param>
              <param name="PO_TYPE" mapped="false"></param>
       </PURCHASEORDER>
       <PURCHASEAMOUNT mapped="false">
              <param name="OBJ_TYPE" mapped="false"></param>
              <param name="OBJ_KEY" mapped="false"></param>
              <param name="OBJ_SYS" mapped="false"></param>
              <param name="OBJ_POS" mapped="false"></param>
              <param name="OBJ_ID" mapped="false"></param>
              <param name="ORVALTYPE" mapped="false"></param>
              <param name="CURR_TYPE" mapped="false"></param>
              <param name="CURRENCY" mapped="false"></param>
              <param name="CURRENCY_ISO" mapped="false"></param>
              <param name="AMT_ORIGIN" mapped="false"></param>
              <param name="AMT_OPEN" mapped="false"></param>
              <param name="AMT_ACTUAL" mapped="false"></param>
              <param name="EXCH_RATE" mapped="false"></param>
              <param name="EXCH_RATE_V" mapped="false"></param>
              <param name="SCHED_LINE" mapped="false"></param>
       </PURCHASEAMOUNT>
```

```
        <CRITERIA mapped="false">
                <param name="ITEMNO_ACC" mapped="false"></param>
                <param name="FIELDNAME" mapped="false"></param>
                <param name="CHARACTER" mapped="false"></param>
        </CRITERIA>
        <VALUEFIELD mapped="false">
                <param name="ITEMNO_ACC" mapped="false"></param>
                <param name="FIELDNAME" mapped="false"></param>
                <param name="CURR_TYPE" mapped="false"></param>
                <param name="CURRENCY" mapped="false"></param>
                <param name="CURRENCY_ISO" mapped="false"></param>
                <param name="AMT_VALCOM" mapped="false"></param>
                <param name="BASE_UOM" mapped="false"></param>
                <param name="BASE_UOM_ISO" mapped="false"></param>
                <param name="QUA_VALCOM" mapped="false"></param>
        </VALUEFIELD>
        <EXTENSION1 mapped="false">
                <param name="FIELD1" mapped="false"></param>
                <param name="FIELD2" mapped="false"></param>
                <param name="FIELD3" mapped="false"></param>
                <param name="FIELD4" mapped="false"></param>
        </EXTENSION1>
</CustomizableMappingFields>
```

In the configuration file, set the `mapped` attribute of the element and its required parameters to TRUE to include them in the input mapping XML. By default, this value is set to FALSE.

Update JavaScript for data massaging

JavaScript for data massaging

The following code block displays a sample JavaScript for data massaging associated to an ERP instance.

```
//This javascript function provides a user to add additional (custom) logic to
manipulate data prior to Create/Park operations in SAP.

/****************************************************************************************
********************
* @param (object) invoice - invoice data. Refer to the installation guide for available
invoice properties.
* @param (object) logger - used to log any information in PCR console. Valid operations
on logger are:
* logger.trace(msg), logger.debug(msg), logger.info(msg), logger.error(msg),
logger.warn(msg)
* For e.g., logger.info("Posting date: " + invoice.PostingDate).
*****************************************************************************************
********************/
function customInvoiceDataMassager(invoice, logger) {
  /*
  if (invoice.PostingDate == null || invoice.PostingDate == undefined ||
invoice.PostingDate.length == 0) {
   invoice.PostingDate = new Date();
  }
  logger.info("Posting date: " + invoice.PostingDate);
  */
}

//This javascript function provides a user to add additional (custom) logic to
manipulate data prior to NonPO Create Invoice operation in SAP.

/****************************************************************************************
********************
* @param (object) invoice - invoice data. Refer to the installation guide for available
invoice properties.
* @param (object) logger - used to log any information in PCR console. Valid operations
on logger are:
* logger.trace(msg), logger.debug(msg), logger.info(msg), logger.error(msg),
logger.warn(msg)
* For e.g., logger.info("Posting date: " + invoice.PostingDate).
*****************************************************************************************
********************/
function customAccountInvoiceDataMassager(invoice, logger) {
  /*
  if (invoice.PostingDate == null || invoice.PostingDate == undefined ||
invoice.PostingDate.length == 0) {
   invoice.PostingDate = new Date();
  }
  logger.info("Posting date: " + invoice.PostingDate);
  */
}
```

You can modify invoice data before the CreateInvoice, ParkInvoice, and CreateNonPOInvoice operations by updating the JavaScript for data massaging.

- To modify invoice data before the CreateInvoice and ParkInvoice operations, edit the section under the following code line.

```
function customInvoiceDataMassager(invoice, logger)
```

The two parameters for customInvoiceDataMassager function–invoice and logger–accept different sets of properties. Properties for invoice must be the same as the input parameters in the XML of CreateInvoice and ParkInvoice operations. For a complete list of properties that can be used, refer to the “Parameter name” column of the table that lists the input parameters of the CreateInvoice and ParkInvoice XML file under ErpAsyncConnector-<version>.jar: CreateInvoice and ParkInvoice.

- To modify invoice data before the CreateNonPOInvoice operations, edit the section under the following code line.

```
function customAccountInvoiceDataMassager(invoice, logger)
```

The two parameters for customAccountInvoiceDataMassager function–invoice and logger–accept different sets of properties. Properties for invoice must be the same as the input parameters in the XML of CreateNonPOInvoice operations. For a complete list of properties that can be used, refer to the “Parameter name” column of the table that lists the input parameters of the CreateNonPOInvoice XML file under ErpAsyncConnector-<version>.jar: CreateNonPOInvoice.

Note The JavaScript standard objects such as Date, Number, Object, Array are set up in the global scope of the script to enable their use.

The following examples display how to edit the customInvoiceDataMassager and customAccountInvoiceDataMassager functions in the JavaScript for three different scenarios.

- To set invoice posting date to current date.

```
if (invoice.PostingDate == null || invoice.PostingDate == undefined ||
invoice.PostingDate.length == 0)

{
   invoice.PostingDate = new Date();
}
```

- To set all invoice lines amounts to positive.

```
var lineItems = invoice. InvoiceLineItemList;
if (lineItems)

{
   for (var i = 0; i < lineItems.length; i++)
   {
      lineItems[i]. ItemAmount = Math.abs(lineItems[i]. ItemAmount);
   }
}
```

- To set GL line items tax code to ‘I0’ for jurisdiction ‘PA0000000’.

```
var glLineItems = invoice. InvoiceGLLineItemList;
if (glLineItems)
{
   for (var i = 0; i < glLineItems.length; i++)
   {
      var taxJurCode = trim(glLineItems[i].TaxJurCode);
      if (taxJurCode == ‘PA0000000’)
      {
         glLineItems[i].TaxCode = ‘I0’;
      }
   }
}
```

SAP user authentication

SAP user authentication

In SAP, create a SAP user of type System and create roles based on details below. Then assign that role to the user with type as System.

1. Add the following TCODES to Role:
   - FB60
   - MIRO
2. Add the following authorization objects to Role:
   1. CPE_SETTIN

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity

   2. S_RFC

Authorization Field	Field Value	Description
ACTVT	16 (Execute)	Activity

Authorization Field	Field Value	Description
RFC_NAME	0002 0012 0015 2105 3006 ACC4 BAPI_COMPANYCODE_GETDETAIL BAPI_COSTCENTER_GETDETAIL1 BAPI_COSTCENTER_GETLIST BAPI_COSTCENTER_GETLIST1 BAPI_GL_ACC_GETLIST BAPI_PROFITCENTER_GETLIST BAPT BREL CJPNCORE DDIF_FIELDINFO_GET LC01 MEWP MLBO MRM_BAPI MSR20_MD OPTB RFC1 RFCH SDIFRUNTIME SDTX SOA1 SOC3 SYST	Remote Function Call (RFC) names
RFC_TYPE	FUGR (Function Group)	RFC type

3. S_TCODE

Authorization Field	Field Value	Description
TCD	FB60 MIRO	Transaction code check

4. S_BTCH_ADM

Authorization Field	Field Value	Description
BTCADMIN	Y	Background admin ID

5. S_BTCH_JOB

Authorization Field	Field Value	Description
JOBACTION	RELE (Release)	Job operations for background jobs
JOBGROUP	*	Summary of jobs for a group

6. S_CTS_ADMI

Authorization Field	Field Value	Description
CTS_ADMFCT	TABL	Admin tasks in change and transport system

7. Table Access:

    1. **7a – dual table objects**: If you are on a SAP release which contains the S_TABU_NAM authorization object, for example, for ECC release 7.x or higher, use the authorizations for table access as shown in the S_TABU_NAM and S_TABU_DIS objects.

        - Authorization Object – S_TABU_NAM

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity

Authorization Field	Field Value	Description
TABLE	ADR6 BKBF BNKA BSEG EKBZ EKKO EKPO LFA1 LFB1 LFBK LFBW LFC3 MARM PRPS RKBP T001 T001W T005 T006A T007S T500W TCURC TCURT TIBAN TKA02 V_T001 WYT3 T001 T042Z BNKA T006A T002	Table name

- Authorization Object – S_TABU_DIS

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
DICBERCLS	FCO SS SC	Table authorization group

2. **7b – single table object**: If you do not have the S_TABU_NAM authorization object, for example, for ECC release lower than 7.x, use the authorizations for table access as shown in S_TABU_DIS object.

- Authorization Object – S_TABU_DIS

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
DICBERCLS	CA FA FB00 FB21 FC FCOR KC MA MCOR PC SA WA FC01 FC12	Table authorization group

8. S_DEVELOP

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
DEVCLASS	/NFM/MM OIA_SI	Package

Authorization Field	Field Value	Description
OBJNAME	/NFM/MM_BAPI_INCOMINGINVOICE_1 /NFM/MM_BAPI_INCOMINGINVOICE_P /NFM/MM_BAPI_PO_GETDETAIL OIA_SI_SAPLMRM_BAPI_I	Object name
OBJTYPE	ENHO	Object type
P_GROUP	*	Authorization group

9. K_CCA

Authorization Field	Field Value	Description
CO_ACTION	0003 (Display Master Data)	Actions for CO-OM
KSTAR	*	Cost element
RESPAREA	*	CO-OM responsibility area

10. K_CSKS

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
KOKRS	*	Controlling area
KOSTL	*	Cost center

11. K_PCA

Authorization Field	Field Value	Description
CO_ACTION	0003 (Display Master Data)	Actions for CO-OM
KSTAR	*	Cost element
RESPAREA	*	CO-OM responsibility area

12. K_PCAR_REP

Authorization Field	Field Value	Description
ACTVT	03 (Display) 27 (Display total records) 28 (Display line items) 29 (Display saved data)	Activity
BUKRS	*	Company code
KSTAR	*	Cost element
PRCTR	*	Profit center

13. K_PCAS_PRC

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
KOKRS	*	Controlling area

14. F_BKPF_BUK

Authorization Field	Field Value	Description
ACTVT	01 (Create or generate) 10 (Post) 77 (Pre-enter)	Activity
BUKRS	*	Company code

15. F_BKPF_KOA

Authorization Field	Field Value	Description
ACTVT	01 (Create or generate) 77 (Pre-enter)	Activity
KOART	*	Account type

16. F_LFA1_APP

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity

Authorization Field	Field Value	Description
APPKZ	*	Customer and vendor master data

17. F_LFA1_GEN

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity

18. F_LFA1_GRP

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
KTOKK	*	Vendor account group

19. F_SKA_BUK

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
BUKRS	*	Company code

20. F_SKA1_KTP

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
KTOPL	*	Chart of accounts

21. M_BEST_BSA

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
BSART	*	Purchasing document type

22. M_BEST_EKG

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity

Authorization Field	Field Value	Description
EKGRP	*	Purchasing group

23. M_BEST_EKO

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
EKORG	*	Purchasing organization

24. M_BEST_WRK

Authorization Field	Field Value	Description
ACTVT	03 (Display)	Activity
WERKS	*	Plant

25. M_RECH_AKZ

Authorization Field	Field Value	Description
ACTVT	02 (Change)	Activity

26. M_RECH_WRK

Authorization Field	Field Value	Description
ACTVT	01 (Create or generate) 03 (Display) 77 (Pre-enter)	Activity
WERKS	*	Plant

27. M_BANF_BSA

Authorization Field	Field Value	Description
ACTVT	03 (Display) 08 (Display Change Documents)	Activity
BSART	*	Purchasing Document Type

28. M_BANF_FRG

Authorization Field	Field Value	Description
FRGCD	10 (Manager) 20 (Controller)	Purchase requisition release code

Custom operations

Custom operations using BAPI

To create and use customized operations using BAPI, complete the following steps.

1. In **[*drive*:]\<Perceptive *Connect Runtime directory*>**, create **BAPIMapping** folder.
2. Write an operation in an XML file and save it in **[*drive*:]\<Perceptive *Connect Runtime directory*>\BAPIMapping** directory. The XML file name should match the corresponding BAPI name. If the BAPI name contains special characters that cannot be used in the XML file name, type the BAPI name as a value for the `BAPIName` attribute in the BAPI element of the XML template.
3. Under **Perceptive Connect,** click **View Bundles** and in the **Bundles** page, click the **Refresh Package Imports** button next to **ErpSyncConnector**.

XML template for custom BAPI operations

Given below is the template for the XML operation file.

```
<?xml version="1.0" encoding="utf-8"?>
<BAPI isActive="true" commitRequired="" functionName="" BAPIName="">
  <IMPORT>
    <params>
      <param name="" BAPIParam="" isMandatory=""></param>
    </params>
    <structures>
      <structure name="" BAPIStruct="">
        <param name="" BAPIParam="" isMandatory=""></param>
      </structure>
    </structures>
       <tables>
         <table name="" BAPITable="" isMandatory="">
           <param name="" BAPIParam="" isMandatory=""></param>
           <param name="" BAPIParam="" isMandatory=""></param>
         </table>
       </tables>
  </IMPORT>
  <EXPORT outputClassName="">
    <params>
       <param name="" BAPIParam=""></param>
    </params>
    <structures>
      <structure name="" BAPIStruct="">
        <param name="" BAPIParam=""></param>
      </structure>
    </structures>
    <tables>
      <table name="" BAPITable="">
        <param name="" BAPIParam=""></param>
      </table>
    </tables>
  </EXPORT>
</BAPI>
```

Remember the following points when you create the XML file.

- `functionName` is the name of the customized synchronous operation. `functionName and outputClassName` should be unique.
- `IMPORT` tag contains the input parameters and `EXPORT` tag contains the output parameters of the operation.
- If you want to edit any record in SAP, set the `commitRequired` to TRUE under `IMPORT` in the XML file.
- If you set the `isMandatory` attribute to TRUE for an input parameter under `IMPORT` in the XML file, then you must provide a value for that parameter while executing the operation.
- If you set the `isMandatory` attribute for a table to TRUE, you should populate at least one row for the table in the SOAP request.
- The input parameters of an operation can contain parameters, structure parameters, and table parameters of a BAPI and the output parameters of the operation can contain parameters, structure parameters, and table parameters of the BAPI.

Given below is an example of custom operation named custGetPODetail, which retrieves the details of a purchase order, based on history and purchase order number.

```
<?xml version="1.0" encoding="utf-8"?>
<BAPI isActive="true" functionName="custGetPODetail">
  <IMPORT>
    <params>
      <param name="purOrder" BAPIParam="PURCHASEORDER" isMandatory="true"></param>
      <param name="History" BAPIParam="HISTORY" isMandatory="false"></param>
    </params>
    <structures />
    <tables />
  </IMPORT>
  <EXPORT outputClassName="PODetails">
    <params />
    <structures>
      <structure name="POHeader" BAPIStruct="PO_HEADER">
        <param name="poNum" BAPIParam="PO_NUMBER"></param>
        <param name="Currency" BAPIParam="CURRENCY"></param>
        <param name="Vendor" BAPIParam="VENDOR"></param>
      </structure>
    </structures>
    <tables>
      <table name="LineItem" BAPITable="PO_ITEMS">
        <param name="lineItemNo" BAPIParam="PO_ITEM"></param>
        <param name="desc" BAPIParam="SHORT_TEXT"></param>
        <param name="Quant" BAPIParam="QUANTITY"></param>
        <param name="unit" BAPIParam="UNIT"></param>
        <param name="TaxCode" BAPIParam="TAX_CODE"></param>
        <param name="netPrice" BAPIParam="NET_PRICE"></param>
      </table>
      <table name="History" BAPITable="PO_ITEM_HISTORY">
        <param name="lineItemNo" BAPIParam="PO_ITEM"></param>
        <param name="RefDoc" BAPIParam="REF_DOC"></param>
        <param name="RefDocYr" BAPIParam="REF_DOC_YR"></param>
        <param name="RefDocItem" BAPIParam="REF_DOC_IT"></param>
      </table>
    </tables>
  </EXPORT>
</BAPI>
```

The following example displays the XML template and parameter list issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:custGetPODetail>
         <ErpId></ErpId>
         <History></History>
         <purOrder></purOrder>
      </syn:custGetPODetail>
   </soapenv:Body>
</soapenv:Envelope>
```

The following example displays the XML template and parameter list issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:custGetPODetailResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <PODetails>
            <POHeader>
               <poNum/>
               <Currency/>
               <Vendor/>
            </POHeader>
         </PODetails>
      </ns1:custGetPODetailResponse>
   </soap:Body>
</soap:Envelope>
```

Custom operations using RFC_READ_TABLE

You can write operations based on RFC_READ_TABLE to customize more searches from SAP. As RFC_READ_TABLE can access any table in SAP, the operations based on it can retrieve live data from any table in SAP.

To create and use customized operations, complete the following steps.

1. In **[*drive*:]\<*Connect Runtime directory*>**, create **BAPIMapping** directory and in **[*drive*:]\PIF\BAPIMapping** directory, create **RFC_READ_TABLE** folder.
2. Write an operation in an XML file and save it in **[*drive*:]\<*Connect Runtime directory*>\BAPIMapping\RFC_READ_TABLE**. The XML file name should match the corresponding operations name.
3. Under **Perceptive Connect,** click **View Bundles** and in the **Bundles** page, click the **Refresh Package Imports** ⟳ button next to **ErpSyncConnector**.

XML template for custom RFC_READ_TABLE operations

Given below is the template for the XML operation file. Input and output parameter names in the XML operation file must be unique and they must not match with any of the existing parameters in the available operations listed in Appendix B: Asynchronous and synchronous operations. You are recommended to prefix the input and output parameters in the XML file with *cust* to avoid errors.

```
<?xml version="1.0" encoding="utf-8"?>
<TABLE name=" " SAPname=" " isActive=" ">
  <INPUT>
    <field name=" " SAPName=" " isMandatory=" "></field>
  </INPUT>
  <OUTPUT>
    <field name=" " SAPName=" "></field>
  </OUTPUT>
  <CONDITIONS clause="C1_AND_(_C2_OR_C3_)">
    <condition id="C1" field=" " operator=" "></condition>
    <condition id="C2" field=" " operator=" "></condition>
    <condition id="C3" field=" " operator=" "></condition>
  </CONDITIONS>
</TABLE>
```

Remember the following points when you create the XML file.

- `SAPName` in the XML is the table name in SAP, which you want to access.
- `name` is the parameter name corresponding to the `SAPName` which you want to display in the operation.
- Data type for all parameters is string.
- Under the `CONDITIONS` tag, you specify the conditional operators for this operation. The table given below shows the list of available operators.

Conditional operator	Description
EQ	Equal to (=)
NE	Not equal to (≠)
G	Greater than (>)
GE	Greater than equal to (>=)
L	Less than (<)
LE	Less than equal to (<=)
SW	Starts with
EW	Ends with
CO	Contain

- Define the conditions and then define the relation between the conditions.

    - You can keep AND and OR while defining the relation between the conditions.
    - You can create conditions with complex relations using braces.
    - Ensure that you use underscores to separate the conditions, operators, and braces.
- While executing the customized synchronous operation, specify the values for ErpId, MaxRowCount, RowSkips, and FiscalYear.
  - The connector operations use Erpld for the SAP connection. For sync calls, ErpId passes as a SOAP parameter. If you do not provide any value for this parameter, the default connection is used. Refer to Set the default SAP connection.
  - MaxRowCount is the maximum number of row counts that the operation includes in the search result. Default is 500 rows.
  - RowSkips is number of rows that the operation skips for pagination.
  - FiscalYear is a mandatory parameter.

Given below is an example of custom operation named custGetDeliveryNotesList, which retrieves the list of delivery notes, based on fiscal year, reference document, and transaction code.

```
<?xml version="1.0" encoding="utf-8"?>
<TABLE name="DeliveryNote" SAPname="MKPF" isActive="true">
  <INPUT>
    <field name="FiscalYear" SAPName="MJAHR" isMandatory="true"></field>
    <field name="Reference" SAPName="XBLNR" isMandatory="true"></field>
    <field name="TransactionCode" SAPName="TCODE2" isMandatory="true"></field>
  </INPUT>
  <OUTPUT>
    <field name="MaterialDocNo" SAPName="MBLNR"></field>
    <field name="FiscalYear" SAPName="MJAHR"></field>
    <field name="TransType" SAPName="VGART"></field>
    <field name="DocumentType" SAPName="BLART"></field>
    <field name="UserName" SAPName="USNAM"></field>
    <field name="Reference" SAPName="XBLNR"></field>
    <field name="TransactionCode" SAPName="TCODE2"></field>
  </OUTPUT>
  <CONDITIONS clause="C1_AND_(_C2_OR_C3_)">
    <condition id="C1" field="FiscalYear" operator="EQ"></condition>
    <condition id="C2" field="Reference" operator="EQ"></condition>
    <condition id="C3" field="TransactionCode" operator="EQ"></condition>
  </CONDITIONS>
</TABLE>
```

The following example displays the XML operation issued for the request.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:syn="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <syn:custGetDeliveryNoteList>
         <ErpId></ErpId>
         <MaxRowCount></MaxRowCount>
         <RowSkips></RowSkips>
         <FiscalYear></FiscalYear>
         <TransactionCode></TransactionCode>
         <Reference></Reference>
      </syn:custGetDeliveryNoteList>
   </soapenv:Body>
</soapenv:Envelope>
```

The following example displays the XML template issued in response to the request.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <ns1:custGetDeliveryNoteListResponse
xmlns:ns1="http://syncoperation.erp.pif.apef.perceptivesoftware.com/">
         <DeliveryNote>
            <TransactionCode/>
            <FiscalYear></FiscalYear>
            <UserName/>
            <TransType> </TransType>
            <MaterialDocNo/>
            <DocumentType> </DocumentType>
            <Reference></Reference>
         </DeliveryNote>
      </ns1:custGetDeliveryNoteResponse>
   </soap:Body>
</soap:Envelope>
```

Create custom wrapper BAPI for fetching payment terms

The following text mentions how to create custom wrapper BAPI for fetching payment terms.

The following procedures are based on an assumption that your SAP system has the Transport Management System configured and running (see transaction STMS).

Import externally created transports into new SAP systems

Navigate to your installation folder and locate the **PerceptiveConnectorForSAPFinancials-<version>.zip** file. Unzip the ZIP file to a suitable location.

Theferore, transports are now available in ***[drive:]\<unzipped_directory>\TransportFiles\PaymentTerms*** directory.

Each transport consists of two files, a '**data**' file and a '**cofile**'. Data files start with the letter 'R' (for example, R900089.EBB) and the cofiles start with the letter 'K' (for example, K900089.EBB), but are otherwise identical in naming. The extension (EBB) is the system which created the transport. Both files are required.

The following steps mention how to import the “ZHYLAND” package. The ZHYLAND package is required if any of the OnBase transports are to be utilized. Many of the transports have a reference that ties it to the ’ZHYLAND’ package. The transport that creates this package should be the first to be imported. The steps mentioned are identical for each OnBase transport.

In order to successfully create the ‘ZHYLAND’ package in another SAP system, perform the following steps.

1. Copy the data and cofile from the directory (for example, R900089.EBB and K900089.EBB) and place inside the appropriate folders in your SAP ‘**trans**’ directory. The data file inside the ‘**data**’ folder and the cofile inside the ‘**cofiles**’ folder. This is typically located at the following location, \\usr\sap\trans.
2. Log into the DEV environment of your SAP system where the package needs to be created.
3. Go to transaction STMS.
4. Click the **Import Overview** icon .
5. Click and select the system designated as the Domain Controller and click the **Display Import Queue** icon.
6. Click **Extras > Other Requests > Add**.
7. Choose the dropdown on the new screen and choose the transport that you manually copied into the ‘**trans**’ directory.
8. Place the cursor on the **Request** that you have just added.
9. Click once on the request to select it and then click the **Import Request** icon .
10. Choose the appropriate **Target Client** and all appropriate items under the **Execution** and **Options** tabs, as directed by the customers basis person. Execute. This starts the import.

    Note After couple of minutes, verify that the ZHYLAND package has been created (transaction code SE80.)

Add functional module

1. Navigate to your installation folder to locate the ‘**data**’ folder and ‘**cofiles**’ folder which are present inside the ***[drive:]\<unzipped_directory>\TransportFiles\PaymentTerms*** directory. Copy the cofile (K900148.EH7) and the data file (R900148.EH7) which are present inside the ‘**data**’ folder and ‘**cofiles**’ folder to target SAP System’s usr\sap\trans respectively.
2. Import the transport request using STMS to the target client (as described above in the Import externally created transports into new SAP systems section).
3. To verify that the import was successful, go to SAP tcode SE37 and display the function module named **ZMSR20_MD_PAYTERMS_GETLIST**.
4. Make sure that the functional module is **Active**, and under the **Attributes** tab, make sure that the **Processing Type** is **Remote-Enabled**.

Create manually

These steps will need to be performed by someone with a developer key to the system.

1. Go to SAP tcode SE80.
2. In the dropdown list, select **Package**. This can either be added to an existing package, or you can create a new one. To create a new one, type in the package name and hit **Enter**. It then asks if you want to create the package, click **Yes**.
3. Enter a description and click the green checkmark.
4. Once created, right-click the **Function Groups** folder and click **Create**. Provide the function group a name and description, then click **Save**.
5. Expand the **Function Groups** folder and right-click **Function Modules**, click **Create**.
6. Type in **ZMSR20_MD_PAYTERMS_GETLIST**, give it a description, then click **Save**.
7. On the **Attributes** tab, make sure that the **Processing Type** is **Remote-Enabled**.
8. On the **Import** tab, copy/paste the parameters from **MSR20_MD_PAYTERMS_GETLIST**.
9. Do the same on the **Export** tab as well as on the **Tables** tab.
10. On the **Source code** tab, select '**Pattern**', enter **MSR20_MD_PAYTERMS_GETLIST** next to **CALL FUNCTION** box, and click the green checkmark.
11. Set the parameters as follows.

```
CALL FUNCTION  ‘MSR20_MD_PAYTERMS_GETLIST’
  EXPORTING
    APPLICATION             = APPLICATION
    PI_LANGU                = PI_LANGU
    PI_ZTERM                = PI_ZTERM
  IMPORTING
    RETURN                  = RETURN
  TABLES
    POT_PAYMENTTERMS        = POT_PAYMENTTERMS
```

12. Click the **Activate** icon.

Index