IBM WebSphere Adapter for SAP Software 7.0.0.0
Quick Start Scenarios

October 1, 2009
This edition applies to version 7 of IBM WebSphere Adapter for SAP Software and to all subsequent releases and modifications unless otherwise indicated in new editions.

© Copyright International Business Machines Corporation 2009. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Table of contents

Chapter 1. Introduction ...........................................................................................................8

Chapter 2. Hardware and software prerequisites .................................................................9

Chapter 3. Installing the adapter ..........................................................................................10

Deployment prerequisites ........................................................................................................10
Configuring the Authentication Alias on the process server .............................................11

Chapter 4. Tutorial 1: Retrieving data from SAP (outbound processing) using the simple BAPI Interface .................................................................14

Business Case ..........................................................................................................................14
Scenario ....................................................................................................................................14
Configuration prerequisites ......................................................................................................15
Configuring the adapter for outbound processing .................................................................16
Setting connection properties for the New External Service wizard ...............................21
Selecting the Business Objects and services to be used with the adapter .................22
Generating Business Object definitions and related artifacts ...........................................25
Deploying the module in the test environment ....................................................................31
Testing the assembled adapter application .........................................................................33
Clearing the sample content .................................................................................................35

Chapter 5. Tutorial 2: Updating data in SAP (outbound processing) using the BAPI Work Unit Interface .................................................................36

Business Case ..........................................................................................................................36
Scenario ....................................................................................................................................36
Configuration prerequisites ......................................................................................................37
Configuring the adapter for outbound processing .................................................................38
Setting connection properties for the External Service wizard ........................................43
Selecting the Business Objects and services to be used with the adapter .....................45
Generating Business Object definitions and related artifacts ...........................................49
Deploying the module in the test environment ....................................................................60
Testing the assembled adapter application .........................................................................62
Clearing the sample content .................................................................................................63

Chapter 6. Tutorial 3: Sending data to an SAP system (outbound processing) using the BAPI ResultSet Interface .................................................................64
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Tutorial: Sending data from an SAP system (inbound processing) using the ALE Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 7</td>
<td>Tutorial 4: Sending data from SAP (INBOUND processing) using BAPI</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Tutorial 5 Sending Structured Query to SAP – Query Outbound Processing</td>
</tr>
<tr>
<td>Chapter 9</td>
<td>Tutorial 6: Sending data from an SAP system (inbound processing) using the ALE Interface</td>
</tr>
</tbody>
</table>
Chapter 14. Troubleshooting AEP ......................................................236

Chapter 15. Tutorial 10: Sending data from the SAP system (inbound processing) and processing it using the ALE-passthrough interface with a Generic IDoc .................................................237

Sending IDoc data From SAP – ALE Inbound Processing ..................................................237
Scenario .........................................................................................................................237
Configuration prerequisites .............................................................................................239
<WPS_INSTALL> represents the WebSphere Process Server installation directory Configuring the adapter for inbound processing..............................................................................240
Setting connection properties for the New External Service wizard..............................245
Selecting the Business Objects and services to be used with the adapter.......................246
Deploying the module in the test environment ..................................................................256
Testing the assembled adapter application .......................................................................258

Chapter 16. Tutorial 11: Sending data to SAP (outbound processing) using the Queued RFC(qRFC) BAPI Interface ..........................259

Configuration prerequisites .............................................................................................259
Configuring the adapter for outbound processing............................................................260
Setting connection properties for the External Service wizard .......................................265
Selecting the Business Objects and services to be used with the adapter .......................267
Generating Business Object definitions and related artifacts ..........................................270
Deploying the module in the test environment ..................................................................276
Testing the assembled adapter application .......................................................................278
Clearing the sample content ............................................................................................279

Chapter 17. Tutorial 12: Sending data from SAP system (INBOUND processing) using qRFC BAPI .........................................................280

Configuration prerequisites .............................................................................................280
Configuring the adapter for inbound processing ..............................................................281
Selecting the Business Objects and services to be used with the adapter .......................288
Generating Business Object definitions and related artifacts .........................................291
Deploying the module in the test environment ..................................................................300
Testing the assembled adapter application .......................................................................301
Chapter 18. Tutorial 13: Generating ALE Audit IDocs per packet (Inbound processing ALE Interface) .................................................................307

Configuration prerequisites .....................................................................................................307
Configuring the adapter for inbound processing ..............................................................308
    Selecting the Business Objects and services to be used with the adapter .......... 315
    Generating Business Object definitions and related artifacts ......................... 319
Deploying the module in the test environment ...............................................................329
Testing the assembled adapter application .......................................................................330

Chapter 19. Tutorial 14: ALE Audit support for Pass through IDocs (Generating ALEAUD per IDoc for ALE Pass through IDocs) ....333

Configuration prerequisites .....................................................................................................333
Configuring the adapter for inbound processing ..............................................................334
    Setting connection properties for the New External Service wizard ............. 339
    Selecting the Business Objects and services to be used with the adapter ....... 341
Deploying the module in the test environment ...............................................................350
Testing the assembled adapter application .......................................................................351

Chapter 20. Tutorial 15: Sending data from SAP (INBOUND processing) using tRFC BAPI ...........................................................................352

Configuration prerequisites .....................................................................................................352
Configuring the adapter for inbound processing ..............................................................353
    Selecting the Business Objects and services to be used with the adapter .......... 360
    Generating Business Object definitions and related artifacts ......................... 363
Deploying the module in the test environment ...............................................................370
Testing the assembled adapter application .......................................................................371

Chapter 21. Tutorial 16: Sending data to an SAP system (OUTBOUND processing) using tRFC BAPI .................................................................373

Configuration prerequisites .....................................................................................................373
Configuring the adapter for outbound processing ........................................................ 374
    Setting connection properties for the New External Service wizard .......... 379
    Selecting the Business Objects and services to be used with the adapter .......... 381
Deploying the module in the test environment ...............................................................392

Notices ..................................................................................................................................394
Chapter 1. Introduction

This document demonstrates how you can use the WebSphere Adapter for SAP Software could be used to provide business integration connectivity to SAP systems.

The BAPI outbound sample demonstrates how the WebSphere Adapter for SAP Software can be used to discover BAPI objects from SAP systems and create Business Objects corresponding to them. The sample shows how to generate a SCA component and, once deployed, how to invoke it to create a Customer object in SAP by invoking the BAPI.

The ALE Inbound sample demonstrates how the WebSphere Adapter for SAP Software can be used to receive events from the SAP EIS by taking a sample IDoc in to consideration. The sample scenario shows how to configure the adapter as an SCA component and, once deployed, how to configure an endpoint to receive a SAP invoked IDoc asynchronous event.

The Structured Query sample scenario demonstrates a query object being created and query data is sent to SAP (Outbound process) and receive a response as a result set based on the query.

Additional scenarios such as BAPI Work Unit, ALE Outbound and ALE inbound for a non-split IDoc are provided explaining about a few scenarios related to ALE and BAPI interfaces.
Chapter 2. **Hardware and software prerequisites**

Please refer to the following link for a listing of hardware and software requirements to run the WebSphere Adapter for SAP Software:

[IBM WebSphere Adapter for SAP Software HW/SW requirements](#)
Chapter 3. Installing the adapter

WebSphere Adapters are packaged along with products like WebSphere Integration Developer and Rational Application Developer. When used with WebSphere Integration Developer, the adapter Resource Archives (RAR) can be located at 
\(<WID\_installation>/ResourceAdapters/SAP_7.0.0.0\) directory.

Note:

The Resource Archive file (RAR) file has support for running the New External Service wizard as well as to process requests coming into the SCA module.

Use the \texttt{CWYAP\_SAPAdapter.rar} file if you want to perform ALE, BAPI, SQI & AEP processing without Local Transaction support. When using the RAR without Local Transaction support, the adapter is in an auto-commit configuration.

Use the \texttt{CWYAP\_SAPAdapter\_Tx.rar} file if you want the container (WebSphere Process Server) to appropriately invoke the commit and rollback methods implemented for the Local Transaction support in the adapter. This is applicable for BAPI transaction and ALE outbound transaction processing scenarios. When using this RAR, the adapter is capable of participating in Local Transactions managed by the container.

The \texttt{CWYAP\_SAPAdapterExt.jar} file available at 
\(<WID\_installation>/ResourceAdapters/SAP_7.0.0.0/\textit{ext}\) contains the JCo 3 provider implementation for use by the adapter.

UNIX® and Windows platforms share the same installed directory and file structure, with the only difference being the directory path designation (forward slash ‘/’ for UNIX, backslash ‘\’ for Windows).

Deployment prerequisites

You must install these products before you can deploy the adapter:

- WebSphere Integration Developer 7.0 (WID)
- WebSphere Process Server 7.0 administrative console

For WebSphere Process Server installation instructions, see the WebSphere Process Server documentation.
After you complete these steps, make sure you have the following information with you to access the SAP application:

- SAP User Name
- SAP Password
- SAP Host name (or IP address)
- SAP System number (usually 00)
- SAP Client number (usually 100)

Obtain the SAP JCo3 from SAP Marketplace at https://websmp101.sap-ag.de/. Please obtain the userID/Password for this URL from your BASIS respectively.

---

**Configuring the Authentication Alias on the process server**

Before you install an application onto WebSphere Process Server, you must create an Authentication Alias for use with your SAP instance. Once an Authentication Alias has been created, other SAP application project modules can use it as well.

Starting the WebSphere Process Server Administrative console:

Open the Business Integration perspective in WebSphere Integration Developer.

Under the Server tab right click on your server instance, for example, **WebSphere Process Server v7.0>Administration** and select **Run administrative console**.
In the administration console of WebSphere Process Server, click **Security** > **Global Security**.

Select Java Authentication and Authorization Service > **J2C authentication data**.

In the **Authentication** column, if an alias named **SAP_Auth_Alias** does not already exist, create it now.

Click **New**. The **General properties** screen appears.
In the **Alias** field, specify **SAP_Auth_Alias** (Remember this authentication alias can be used while generating inbound\outbound services using the adapter).

Specify the User ID and password that are required to connect to the SAP system.

Click **OK**.

Click **Save**.
Chapter 4. **Tutorial 1: Retrieving data from SAP (outbound processing) using the simple BAPI Interface**

**Business Case**

A Customer Relationship Officer needs to retrieve a customer’s information from the SAP server after being notified of a customer complaint raised on the Customer Services Portal.

**Scenario**

The following scenario illustrates a simple BAPI outbound processing using Synchronous RFC calls.

*Figure: Scenario illustrating simple BAPI outbound processing*
1. The Customer Services portal notifies the Officer of a new complaint.

2. The Officer logs into the Customer Services portal and requests details of the customer.

3. The Customer Services portal will in turn invokes an SCA module using the WebSphere Adapter for SAP Software to execute the BAPI_CUSTOMER_GETDETAIL BAPI.

   i. The adapter module receives a request from the client application in the form of a BAPI_CUSTOMER_GETDETAIL Business Object. The adapter sends this data to SAP server.

   ii. The adapter then converts the data in the Business Object into a native BAPI object and sends it to the SAP server.

4. The SAP server responds with the details of the customer in native object format.

5. The adapter handles the response from SAP, converts it into a Business Object format as required by the client application and returns the response.

6. The Customer Services portal returns the response to the officer.

---

**Configuration prerequisites**

You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector interface is an external dependency that the adapter requires in order to connect to the SAP systems. The adapter uses SAP JCo to call to the SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library interface to the imported project. You must copy all external libraries and JAR files must first be copied to the appropriate locations on WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.
Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add

${WAS_INSTALL_ROOT}/lib/sapjco3.jar file to

WAS_SERVER_ONLY_server_region_classpath

The sapjco3.jar is required to run the New External Service wizard.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

Copy `CWYAP_SAPAdapterExt.jar` to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add

${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar to

WAS_SERVER_ONLY_server_region_classpath

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the New External Service wizard to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually Business Integration.

Start the New External Service wizard by choosing: File -> New -> External

1. Select Adapters > SAP in the Select the Service Type of Registry screen and click Next.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support** (IBM: 7.0.0.0) in the Select an Adapter screen and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the New External Service wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR** File screen and click **Next**.

![Figure: Import a RAR file screen](image-url)
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

5. Click **Next**.

6. In the **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.
Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

User name
Password
Host name
System number
SAP Client connection

Click Select to change the default Language code from English
Use the drop down option to change the default Code page from 1100.

Select BAPI as the SAP Interface name.

Click Next.

![Figure: Select BAPI as the interface](image)

Selecting the Business Objects and services to be used with the adapter

1. In the Find objects in the Enterprise System screen, click on RFC node. Then click the button to enter a filter.
2. Enter BAPI_CUSTOMER_GET* (the name of the BAPI in SAP plus an asterisk as a wild card character) in the Filter Properties for ‘RFC’ screen.
3. Click **OK**.

4. Expand the **RFC** node.

5. Select the **BAPI_CUSTOMER_GETDETAIL** from the metadata tree.

6. Click the **>>** button.

7. A popup will appear containing the Configuration properties for the **BAPI_CUSTOMER_GETDETAIL** object.

8. Check the **Use SAP field names to generate attributes names** checkbox if you want the Business Object attribute names to be generated using SAP field Names.

*Figure: Retrieved BAPIs based on search criteria*
9. You can choose to create attributes in the Business Object for any optional parameter in the BAPI.

Click **OK**.

![Configuration Properties for 'BAPI_CUSTOMER_GETDETAIL'](image)

**Figure: Setting configuration parameters for the BAPI selected**

10. Click **Next**.

**Generating Business Object definitions and related artifacts**

In the Specify Composite Properties screen, you can enter a business object folder name where the Business Objects should be created instead of the default location.

Enter **bodefs** as the folder name.

Check the Generate BAPI Business Objects within a wrapper.

Enter **Cust_GetDetail** in the Business object for service operations field.

Under **Service Operations**, click **Add** and add the **Retrieve** operation.
Click Next.

In the Service Generation and Deployment Properties screen, enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select the Module radio button and click Next.

In the New Business Integration Project screen, select the Module option and click Next.

In the New Module screen, type BAPI_CUST_GETDETAIL in the Module Name field, and then click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_CUST_GETDETAIL by selecting Add and Remove Projects.

The project BAPI_CUST_GETDETAILApp will be listed under Available projects.
Click on the project name and Add it to the list of **Configured Projects**. Click **Finish** to deploy the SCA module.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. After you have deployed the module to the server, right click the module BAPI_CUST_GETDETAIL in the Projects view and select Test > Test Module from the pop-up menu.

2. Enter values in the Test Client as shown in the following figure.

![Figure: Entering the Input values](image)

3. Click the Continue button.

When the Select Deployment screen appears, select the WebSphere Process Server to which you added the project and click the Finish button.

4. If security is enabled, type in the username and password in the popup User Login screen that appears and click OK.
5. Check the output of the service in the test client.

6. Check that the data retrieved in the Test Client matches your EIS data.

7. Login to the SAP GUI using the credentials for the given SAP server

8. Open the BAPI Explorer (tcode – se37)
9. Enter the BAPI name BAPI_CUSTOMER_GETDETAIL in the Function Module field,

10. Press F8 to execute the BAPI.

11. Enter the values provided in step (2) above for Test Function Module: Initial screen, under Import parameters

12. Press F8 to Execute the BAPI using the values that are entered.

13. Check that the values in Export Parameters in the Test Function Module: Result screen and match the values returned in step (5) above

To read the documentation pertaining to any BAPI.

1. Open the BAPI Explorer using tcode – se37.

2. Enter the name of the BAPI in the Function Builder: Initial Screen.

3. Click Display.

4. Click Function Module Documentation or press Ctrl + F4

---

Clearing the sample content

No clean up is required after this tutorial.
Chapter 5. Tutorial 2: Updating data in SAP (outbound processing) using the BAPI Work Unit Interface

Business Case

A Manager has to update an employee’s address in an SAP system when informed of a change.

Scenario

The following scenario illustrates outbound processing of BAPI units of work. A BAPI work unit consists of a set of BAPIs that are processed in sequence to complete a task.

1. The HRMS portal notifies the Manager with the details of the address change of an employee.
2. The Officer logs in to the administrative console of the HRMS Portal and submits the address change request for the employee.

3. The HRMS portal will invoke the SCA Import configured to update an employee record using the WebSphere Adapter for SAP Software. This is accomplished by calling the three BAPIs, in the same work unit sequentially:

   BAPI_EMPLOYEE_ENQUEUE
   BAPI_EMPLOYEE_DEQUEUE
   BAPI_EMPLOYEE_CHANGE

4. The first BAPI locks the employee record. The second updates the record, and the third approves the update.

   **Note**: The advantage of using the BAPI Work Unit interface is that the client application can request the employee record change with a single call, though the work unit consists of three separate functions. Also, the BAPI Work Unit interface can be used if your SAP system requires that certain BAPIs are to be processed in a specific sequence for a business flow to complete correctly.

5. The adapter receives a request from the client application as a wrapper Business Object that contains these three child BAPI objects.

   BAPI_ADDRESSEMP_ENQUEUE
   BAPI_ADDRESSEMP_DEQUEUE
   BAPI_ADDRESSEMP_CHANGE

6. The adapter sends this data to the SAP system which responds with the details of the three BAPIs.

7. The adapter handles the SAP response, converts it into a Business Object format as required by the client application and returns it.

8. The HRMS portal returns the appropriate response to the Manager.

---

**Configuration prerequisites**

You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.
Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory. (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/sapjco3.jar` to WAS_SERVER_ONLY_server_region_classpath.

The sapjco3.jar is required to run the New External Service wizard.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar` to WAS_SERVER_ONLY_server_region_classpath.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: File-> New -> External Service.

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click Next.
2. Select the IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0) from the Select an Adapter screen and click Next.
Figure: Select an Adapter screen

**Note**: If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node.

3. Specify a Connector Project name in the Import a RAR File screen and proceed by clicking on **Next**.


Figure: Import a RAR file screen
4. In the **Locate the Required Files and Libraries** Screen, provide the location of sapjco3.jar and sapjco3.dll or libsapjco3.so files.

5. Click **Next**.

6. In the **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.
Setting connection properties for the External Service wizard

You must provide the following information in the Discovery Configuration screen:

User name
Password
Host name
System number
SAP Client connection
Click Select to change the default Language code from English
Use the drop down option to change the default Code page from 1100.

Select **BAPI Work Unit** as the SAP Interface name.

Click **Next**.
Selecting the Business Objects and services to be used with the adapter

1. In the **Find objects in the Enterprise System** screen, click expand RFC node. Then click on the button.

![Find Objects in the Enterprise System](image)

*Figure: Object Discovery and Selection*

2. Enter BAPI_EMPLOYEE_ENQUEUE (the name of the BAPI in SAP plus an asterisk as a wild card character) in the Filter Properties for Discover by Name screen.
3. Click **OK**.

4. Expand the **RFC** node.
5. Select the BAPI_EMPLOYEE_ENQUEUE.

6. Click the button.

7. A popup will appear containing the Configuration properties for the BAPI_EMPLOYEE_ENQUEUE object.

Check the **Use SAP field names to generate attributes names checkbox** if you want the Business Object attribute names to be generated using SAP field Names.
8. You can choose to create attributes in the Business Object for any optional parameter in the BAPI.

9. Click **OK**.

![Figure: Setting configuration parameters for the BAPIs selected](image)

10. Click **OK**.

11. Repeat steps 1 to 6 for the following BAPIs –

   **BAPI_ADDRESSEMP_CHANGE**

   **BAPI_EMPLOYEE_DEQUEUE**
12. Click **Next**.

**Generating Business Object definitions and related artifacts**

In the Specify Composite Properties screen, enter the Business Object name for service operations as **BAPI_EMPLOYEE_ENQUEUE**.

Under **Service Operations**, click Add and add the **Create** as an operation
Associate the Business Object BAPI_ADDRESSSEMP_REQUEST to the Create operation by clicking Add under Sequence of RFC functions for the selected operation.

The Specify Composite Properties screen will now look like the following screen.
Under **Service Operations**, click Add and add the **Update** as an operation.
Associate the Business Object BAPI_ADDRESSSEMP_CHANGE to the Update operation by clicking Add under Sequence of RFC functions for the selected operation.
The **Specify Composite Properties** screen will now look like –

Under **Service Operations**, click Add and add the **Retrieve** as an Operation.
Associate the Business Object BAPI_ADDRESSSEMP_APPROVE to the Retrieve operation by clicking Add under **Sequence of RFC functions for the selected operation**.

Optionally, enter the Business Object folder name for the folder in which the business objects are created. **bodefs** has been used as the folder name for this example.

The **Specify Composite Properties** screen will now look like –
To verify the selections, click any service operation selected and its corresponding Business Object should be displayed automatically in the text box below it.

Click Next.

In the Service Generation and Deployment Configuration screen, enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login into the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.

In the New Module screen, type BAPIADDRESSUPDATE_WORKUNIT in the Module Name field, and click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_ADDRESS_UPDATE_WORKUNIT by selecting Add and Remove Projects.

The project BAPI_ADDRESS_UPDATE_WORKUNIT App will be listed under Available projects.
The project you added will appear under the Configured projects. Add the SCA module to the server by clicking Finish.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer’s Test Client.

1. Once the module is deployed to the Server, right click the module BAPI_ADDRESS_UPDATE_WORKUNIT from the Projects view and select Test > Test Module from the pop-up menu.

2. Enter values in the Test Client as shown below.

3. Navigate to the SapBapiCustomerCreatepwrreg node of Initial request parameters field and expand it if it is not expanded. Enter the customer number 0000000001.

4. Click the Continue button.

When the Select Deployment screen appears, select the WebSphere Process Server instance to which you added the project and click the Finish button.

5. If security is enabled, type in the username and password in the popup User Login screen that appears and click OK.
6. Check the output of the service, and check the data in the EIS to ensure it matches expected values.

---

**Clearing the sample content**

No clean up is required after this tutorial.
Chapter 6. **Tutorial 3: Sending data to an SAP system (outbound processing) using the BAPI ResultSet Interface**

**Business Case**

The Audit team requires details of all customers of the company. Every customer record has a unique customer number which should be used to lookup their details.

**Scenario**

The following scenario illustrates simple BAPI outbound processing using Synchronous RFC calls.

1. The Audit team logs in to the Customer Services portal and accesses the details of all the customers.
2. The Customer Services portal invokes the SCA Import configured to retrieve information of all the customers with WebSphere Adapter for SAP. This is accomplished by using the BAPI Result Set interface.

BAPI Result Sets use the GetList and GetDetail functions to retrieve an array of data from the SAP server. The information returned from the GetList function is used as input to the GetDetail function.

Here, the set of customers are retrieved using BAPI_CUSTOMER_GETLIST, which acts as a query BAPI, and BAPI_CUSTOMER_GETDETAIL, which acts as the result BAPI.

The BAPIs perform the following steps:

i. BAPI_CUSTOMER_GETLIST call returns the list of Customer Numbers

ii. Each Customer Number is mapped dynamically to the Business Object for BAPI_CUSTOMER_GETDETAIL.

iii. BAPI_CUSTOMER_GETDETAIL is processed multiple times for every Customer Number, so that an array of customer information is returned

3. The adapter receives a request from the client application in the form of a BAPI Result Set Business Object. The adapter sends this data to SAP server.

4. The SAP server responds with the details of the customer records.

5. The adapter handles the response from SAP, converts it back to a Business Object format as required by the client application and is returned.

6. The Customer Services portal returns the response to the Auditing Personnel.

---

**Configuration prerequisites**

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP's native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory.

(If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).
When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the 
<WPS_INSTALL>/lib directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the <WPS_INSTALL>/lib directory.

When working with WebSphere Process Server on z/OS, add 
${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to
WAS_SERVER_ONLY_server_region_classpath

The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Configuring the adapter for outbound processing

Run the New External Service wizard to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually Business Integration

Start the New External Service wizard by choosing: File -> New -> External Service

1. Select Adapters > SAP from the Select the Service Type of Registry screen and click Next.
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.
Note: If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node

3. Specify the Connector Project name in the Import a RAR File screen and proceed by clicking Next.
Figure: Import a RAR file screen
4. In the Locate the Required Files and Libraries screen, provide the locations of sapjco3.jar and sapjco3.dll or libsapjco3.so files.

5. Click Next.
6. In the **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.

**Setting connection properties for the New External Service wizard**

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
SAP Client connection

Click Select to change the default Language code from English

Use the drop down option to change the default Code page from 1100.

Select BAPI result set as the SAP Interface name.

Click Next.

![New External Service window]

Figure: Select BAPI Result Set as the interface

Selecting the Business Objects and services to be used with the adapter

1. In the Find objects in the Enterprise System screen, expand the RFC node. Then click the button.
Figure: Object Discovery and Selection
Enter BAPI_CUSTOMER_GET* (the name of the BAPI in SAP and an asterisk as a wild card character) in the Filter Properties for ‘RFC’ screen.

Figure: Filter Properties for RFC

Click OK.

Expand the RFC node.
Select both the BAPI_CUSTOMER_GETDETAIL and BAPI_CUSTOMER_GETLIST BAPIs’ from the list of discovered BAPIs’.

Click the button.

A popup will appear containing the Configuration properties for the BAPI_CUSTOMER_GETDETAIL object.

Check the **Use SAP filed names to generate attributes names checkbox** if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI.
Click OK.

![Configuration Properties](image)

*Figure: Setting configuration parameters for the BAPIs selected*

Click Next.

**Generating Business Object definitions and related artifacts**

In the Specify Composite Properties screen,

a) Enter the name of the Business Object as **CustomerDetail**.

b) Map the field in **Query BAPI** to BAPI_CUSTOMER_GETLIST. The Result BAPI field will be mapped to BAPI_CUSTOMER_GETDETAIL.
c) Click **Add** button. This brings up a pop-up screen, Add/Edit properties in which both the BAPIs required for the result set are displayed.

Click the Select button corresponding to SapBapiCustomerGetDetail and choose SapBapiCustomerGetDetail/CustomerToBeRequired.

Similarly click the Select button corresponding to SapBapiCustomerGetList and choose SapBapiCustomerGetList/SapAddressData/CustomerNumber1.

Click Finish.
d) Enter the folder name for the generated Business Object as **bodefs**.
Click **Next**.

In the **Service Generation and Deployment Configuration** screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click on Next.

In the New Module screen, type BAPI_GET_DETAIL_RESULTSET in the Module Name field, and then click Finish.
Click Finish on the Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_GET_DETAIL_RESULTSET by selecting Add and Remove Projects.

The project BAPI_GET_DETAIL_RESULTSETApp will be listed under Available projects.

After adding the project, the added project should appear under the Configured projects. Add the SCA module to the server. Click Finish.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Right click the module “BAPI_GET_DETAIL_RESULTSET” and select Test > Test Module from the pop-up menu.

2. Enter values as shown in the figure below:

3. Right click SapBapiCustomerGetDetail and select AddElements and enter 1 as the value in the popup screen.

   a) Enter the following values for the added element SapBapiCustomerGetDetail[0]

   CustomerToBeRequired=0000000001 DistributionChannel=01 Division=01 SalesOrganization=0001

   b) Under QueryBO set the following data

   MaximumNumberOfCustomers=3

   c) In QueryBO we have another element with name SapIdRange. Right click SapIdRange and select AddElements and enter 1 as the value in the popup screen.
InclusionExclusionCriterionSignForRangeTables= E SelectionOperatorOptionForRangeTables= 
EQ CustomerNumber1=10     CustomerNumber2217378=25

d) Under the QueryBO we also have SapAddressData. Right click SapAddressData and select AddElements and enter 1.

CustomerNumber=0000000001

4. Click the **Continue** button.

When the **Select Deployment** screen appears, select the WebSphere Process Server instance to which you added the project and click the **Finish** button.

5. If security is enabled, type in the username and password in the popup User Login screen that appears and click **OK**.
6. Check the output of the service, and check the data in the EIS to ensure it matches expected values.

**Clearing the sample content**

No clean up is required after this tutorial.
Chapter 7. Tutorial 4: Sending data from SAP (INBOUND processing) using BAPI

Business Case

The Sales department of an organization stores the information related to new customers in their SAP system and later updates the Accounts department after verifying the details.

Scenario

The following scenario illustrates BAPI inbound processing synchronously (in which both the client application and the adapter must be available during processing). In synchronous processing, the client application waits for a response from the adapter.

Figure: Scenario illustrating simple BAPI inbound processing

1. The Sales department of an organization stores the details of a new customer in SAP system.
2. The WebSphere Adapter for SAP is configured for synchronous BAPI Inbound processing. The adapter starts event listeners, which listen for RFC-enabled function events (specified with the RFCProgramID property) from the SAP server.

3. The customer details are to be fetched using BAPI_CUSTOMER_GETDETAIL by providing a customer number.

4. This RFC enabled function event is pushed to the adapter by the way of the event listeners. The adapter resolves the operation and Business Object name using the received RFC-enabled function name.

5. The adapter sends the Business Object to the configured end-point in a synchronous manner. The end point might be any application that updates the accounts department.

6. The adapter receives the response Business Object from the end point.

7. The adapter maps the response Business Object to an RFC-enabled function and returns it to the SAP Server

**Configuration prerequisites**

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath`
The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

### Configuring the adapter for inbound processing

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click on **Next**.
Figure: Select the Service Type or Registry screen

Creates a service to access the SAP server using WebSphere® Adapter for SAP Software. You can create service-oriented integrated processes, which can interact and exchange information with the SAP Server, without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on the SAPEIS. During inbound processing, these services use the adapter to receive events from the SAP EIS.
2. Select the IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0) from the Select an Adapter screen and click Next.

![Select an Adapter screen]

**Figure: Select an Adapter screen**

**Note:** If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node.
3. Specify a Connector Project name in the **Import a RAR File** screen and proceed by clicking **Next**.

![Image: Import a RAR file screen]

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** Screen, provide the locations of the sapjco3.jar and sapjco3.dll or libsapjco3.so files

5. Click **Next**.

6. In the **Select the Processing Direction** screen, select the **Inbound** radio button, then click **Next**.
Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

- User name
- Password
- Host name
- System number
- SAP Client connection

Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.
Select BAPI as the SAP Interface name.

Then click **Next**.

---

**Figure: Select BAPI as the interface**

**Selecting the Business Objects and services to be used with the adapter**

Under **Find Objects in the Enterprise System**, click on RFC node. Then click the **Next** button.
Enter BAPI_CUSTOMER_GET* (the name of the BAPI in SAP and an asterisk as a wild card character) in the **Filter Properties for ‘RFC’** screen.

Click **OK**.

Expand **RFC** node.
Select the BAPI_CUSTOMER_GETDETAIL from metadata tree.

Click the button.

A popup will appear containing the Configuration properties for the BAPI_CUSTOMER_GETDETAIL object.

Check the Use SAP filed names to generate attributes names checkbox if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI.
Click OK to go back to Find Objects in the Enterprise System screen.

Figure: Setting configuration parameters for the BAPI selected

Click Next.

Generating Business Object definitions and related artifacts

In the Specify Composite Properties screen, associate the RFC-enabled function name with an end-point operation Create.

Enter the name of the folder where the Business Objects are to be created as bodefs.
Click **Next**.

In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties. Click **Next**.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP.

Enter the RFCProgramID (as shown in figure). This must have been previously configured in the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.

In the New Module screen, type BAPI_CUST_GETDETAIL_IN in the Module Name field, and then click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the “BAPI_CUST_GETDETAIL_IN” SCA module, and double click the Assembly Diagram. The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button of the required java component from the Palette.
Click and drag the Java component to add the new component to the Assembly Diagram screen.

Add a Wire between the **SAPInboundInterface** and the Java component.

![Diagram](image)

*Figure: BAPI Inbound interface being wired to a target Component (end-point)*

1. In the Add Wire screen, click OK.

![Add Wire Screen](image)

*Figure: Add Wire Screen*

2. Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.

```java
/*
 * Method generated to support implementation of operation "emitCreateAfterImageSapBapiCustomerGetDetailWrapper" defined for WSDL port type
 * named "SAPInboundInterface".
 * 
 * The presence of `com.sap.DataObject` as the return type and/or as a parameter
 * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */

public DataObject emitCreateAfterImageSapBapiCustomerGetDetailWrapper(
    DataObject emitCreateAfterImageSapBapiCustomerGetDetailWrapperInput) {
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSapBapiCustomerGetDetailWrapperInput));
    } catch (Exception e) {
        e.printStackTrace();
    }

    return emitCreateAfterImageSapBapiCustomerGetDetailWrapperInput;
}
```

Save the Java file and the assembly diagram.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_CUST_GETDETAIL_IN by selecting Add and Remove Projects.

The project BAPI_CUST_GETDETAIL_INApp will be listed under Available projects.

The project that you added should appear under the Configured projects. Add the SCA module to the server by clicking Finish.
Testing the assembled adapter application

Launch the SAP GUI.

Start the Transaction SE37.

Enter the BAPI name as BAPI_CUSTOMER_GETDETAIL and execute the BAPI

Enter the RFC target sys as ‘JCO3RFCSERVER’ (as this is the RFC Program ID used when we configured the module)

Enter the other data as shown in the figure below

Execute the BAPI
In the console of WebSphere Integration Developer, you could see the BAPI_CUSTOMER_GETDETAIL Business Object dump.

Clearing the sample content

No clean up is required after this tutorial.
Chapter 8. Tutorial 5 Sending Structured Query to SAP – Query Outbound Processing

This tutorial demonstrates how the WebSphere Adapter for SAP Software’s QISS interface can be used to retrieve data from tables directly (in this example, we will retrieve data from SAP’s table Kna1).

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the <WPS_INSTALL>/bin directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at <WID_INSTALL_DIR>/runtimes/bi_v7).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the <WPS_INSTALL>/lib directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the <WPS_INSTALL>/lib directory.

When working with WebSphere Process Server on z/OS, add ${WAS_INSTALL_ROOT}/lib to WAS_SERVER_ONLY_server_region_classpath

The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.
Copy CWYAP_SAPAdapterExt.jar to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add

`${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar` to

`WAS_SERVER_ONLY_server_region_classpath`

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click **Next**.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** Screen, provide the locations of the sapjco3.jar and sapjco3.dll or libsapjco3.so files.

5. Click **Next**.
6. In the **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.

**Setting connection properties for the New External Service wizard**

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
SAP Client connection

Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select Query interface for SAP Software (QISS) as the SAP Interface name.

Click **Next**.

*Figure: Select QISS as the interface*
Selecting the Business Objects and services to be used with the adapter

In the Find objects in the Enterprise System screen, click the QISS node. Then click the button.

*Figure: Object Discovery and Selection*
Enter KNA1 (the name of the QISS object) in the **Filter Properties for ‘QISS’** screen.

![Filter Properties for 'QISS'](image)

*Figure: Filter Properties for RFC*

Click **OK**.

Expand the **QISS** node.
Click the button to select KNA1 into Selected Objects.
Accept the defaults and click **OK**.

Click the **QISS** node and then the button again.

Repeat Steps 4-7 above for the pattern **ADRC**.

In Specify the Configuration Properties for **ADRC** -

Under **Select a Parent Table** for Table ADRC, select KNA1 from the drop down list.

Under the Map the primary key columns to the parent-table foreign key reference columns section, choose the following from the drop down list:

<table>
<thead>
<tr>
<th>CLIENT</th>
<th>ADRNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Select the defaults for Choose columns to be selected in your query.

Click **OK**.

Click **Next**.
Generating Business Object definitions and related artifacts

In the Specify Composite Properties screen, use the default value for the Business object namespace.

Name the relative folder name for generated Business Objects as **bodefs**.

![Specify Composite Properties](image)

*Figure: Specify Composite Properties*

Click **Next**

In **Specify the Service Generation and Deployment Configuration** screen enter the connection properties and deployment properties.
Specify the Service Generation and Deployment Properties

Service Operations
To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations.

Deployment Properties
How do you want to specify the security credentials?
- Using an existing JAAS alias (recommended)
  A Java Authentication and Authorization Services (JAAS) alias is the preferred method.
  J2C authentication data entry:

- Using security properties from the managed connection factory
  The properties will be stored as plain text; no encryption is used.
  User name: * smandur
  Password: * ********

- Other
  Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

The quality of service to join the transaction provides a higher degree of data integrity, especially in the event of a failure. As the adapter only supports local transactions, it must be the only one phase commit capable resource in the transaction. More...

- Join the transaction (recommended)

Deploy connector project:
- With module for use by single application

Specify the settings used to connect to SAP® Software at run time:
Connection settings:

SAP system connection information
- Use load balancing
  To use load balancing, specify the load balancing properties in the Additional connection configuration panel under the Advanced tab.
  Host name: * cwd31.svw.bm.com
  System number: 01
  Client: 100
  Language code: EN (English)
  Code page: 1100

Advanced >>
**Note:** You can enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system.

Click **Next**.

In the **Specify the Location Properties** screen, click the **New** button next to the Module field to create a new module.

In the **New Business Integration Project** screen, select **Module** radio button and click **Next**.

In the New Module screen, type **QISSKna1AdrcSample** in the Module Name field, and then click **Finish**.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer's Test Client.

Right click on your server node in the Server tab and add the module QISSKna1AdrcSample by selecting Add and Remove Projects.

The project QISSKna1AdrcSampleApp will be listed under Available projects.

After adding the project, the added project will appear under the Configured projects. Add the SCA module to the server by clicking Finish.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer’s Test Client.

Once the module is deployed to the Server, right click the module QISSKna1AdrcSample from the Projects view and select Test > Test Module from the pop-up menu.

Enter the values for the input Business Objects as below –

Select the Operation as retrieveallSapkna1

Set the value of CustomerNumber1 as 0000000001
Add a child object ADRC by right clicking on SapAdrc Business Object and adding 1 element by clicking Add Elements.

Click the Continue button.

When the Select Deployment screen appears, select the WebSphere Process Server instance to which you added the project and click the Finish button.

If security is enabled, type in the username and password in the popup User Login screen that appears and click OK.
Check the output of the service in the test client

Check that the data in the EIS matches the above output –
Login to the SAP GUI using the credentials for the given SAP server

Start the Transaction SE16N

Enter the table name as **KNA1**, Press Enter and Execute (F8). Double clicking on **CustomerNumber ‘1’** will display the Customer details.

Repeat the (c) above for table ‘ADRC’.

---

**Clearing the sample content**

There is no clean up required after following this tutorial.
Chapter 9. Tutorial 6: Sending data from an SAP system (inbound processing) using the ALE Interface

Sending IDoc data From SAP – ALE Inbound Processing

This tutorial demonstrates how to use the New External Service to generate Business Objects based on the IDoc, and create an SCA module using the WebSphere Adapter for SAP Software 7.0.0.0 and deploy the module to the test environment of WebSphere Integration Developer 7.0.

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP's native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the <WPS_INSTALL>/bin directory
(If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at <WID_INSTALL_DIR>/runtimes/bi_v7).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the <WPS_INSTALL>/lib directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the <WPS_INSTALL>/lib directory.

When working with WebSphere Process Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to
WAS_SERVER_ONLY_server_region_classpath
The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Configuring the adapter for inbound processing

Run the New External Service wizard to generate Business Objects, Services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually Business Integration

Start the New External Service wizard by choosing: File-> New -> External Service

1. Select Adapters > SAP from the Select the Service Type of Registry screen and click Next.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note**: If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify the Connector Project name in the **Import a RAR File** screen and click **Next**.

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

![Locate the Required Files and Libraries screen](image)

*Figure: Locate the required files and Libraries screen*

5. Click **Next**.

6. In the **Select the Processing Direction** screen, select **Inbound** radio button, then click **Next**.
Setting connection properties for the New External Service wizard

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
- Click Select to change the default Language code from English
- Use the drop down option to change the default Code page from 1100.

Select **ALE** as the SAP Interface name.

*Figure: Select ALE as the interface*
Selecting the Business Objects and services to be used with the adapter

Under Find Objects in the Enterprise System, expand the ALE node and click the Discover IDoc From System.

Click the button.

Enter Alereq01 (the name of the ALE in SAP system) in the Filter Properties for Discover IDoc From System screen.
Click **OK**.

Expand the Discover IDoc From System node.

Select **ALEREQ01** and click the **button**
In the Configuration Parameters screen, choose the default values and click **OK**.
Figure: Setting configuration parameters for the ALE selected

ALEREQ01 has now been added to the list of Business Objects to be imported.
Click Next.

Generating Business Object definitions and related artifacts

In the Specify Composite Properties screen –

Select the Service operation as Create

Add MessageType=ALEREQ; MessageCode=; MessageFunction=; as IDoc Identifiers for the service operation by clicking on Add button
Enter `bodefs` as the name of the relative folder for the generated Business Object.

Figure: Specify Composite properties

Click **Next**

In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
NOTE: You can enter an Authentication Alias already previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login into the SAP.

Enter the RFCProgramID (as shown in figure). This must have been already configured in the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.

1. In the New Module screen, type ALEIN_ALEREQ01 in the Module Name field, and then click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the BAPI_CUST_GETDETAIL_IN SCA module, and double click the Assembly Diagram. The Assembly Diagram screen appears with the module’s Export component in view.

1. To create a new component, click the button of the Java component from the Palette.

Click and drag the Java component to add the new component to the Assembly Diagram screen.
Add a Wire between the **SAPInboundInterface** and the Java component.

In the Add Wire screen, click OK.

Right-click on the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.

```java
/**
   * Method generated to support implementation of operation "emitCreateAfterImageSapAlereq01" defined for WSDL port type
   * named "SAPInboundInterface".
   *
   * The presence of common.sdo.DataObject as the return type and/or as a parameter
   * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
   * on the type of input, output and fault(s).
   */

public void emitCreateAfterImageSapAlereq01(
    DataObject emitCreateAfterImageSapAlereq01Input) {
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSapAlereq01Input));
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Save the Java file and the assembly diagram.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_CUST_GETDETAIL_IN by selecting Add and Remove Projects.

The project BAPI_CUST_GETDETAIL_INApp will be listed under Available projects.

The project that you added should appear under the Configured projects. Add the SCA module to the server by clicking Finish.
Testing the assembled adapter application

Launch the SAP GUI.

Start the Transaction WE19.

Choose the radio button **Existing IDoc**

Select an existing IDoc ALEREQ01 that you want to send

Set appropriate values in IDoc.

Click Standard Outbound Processing button

Select **Continue** in the pop-up box

This creates an ALE inbound event for the ALE inbound application deployed earlier.

In the console of WebSphere Integration Developer, you will see the ALEREQ01 Business Object printed (as we entered a print statement in the Component implementation above).
Chapter 10. **Tutorial 7: Sending data to an SAP system (Outbound processing) using the ALE Interface**

**Sending IDoc data to SAP – ALE Outbound Processing**

This tutorial demonstrates how to use the **New External Service** wizard to generate Business Objects based on the IDoc, and create an SCA module using WebSphere Adapter for SAP Software 7.0.0.0 deploy the module on to the WebServer Process Server to finally create IDoc in SAP system. This tutorial uses the Basic IDoc **ALEREQ01**.

**Scenario**

The scenario explains how ALE outbound processing can be used in a real business case.

A user requests for information regarding a material through a website.

The website requires you to provide an email id to send the information to.

The whole communication model is asynchronous, and hence ALE interface is used.
**Note:** The tutorial only covers the part of the scenario marked in red colored rectangle in the above picture.

1. The user submit request for a Material information in the Customer Services Portal using Material number

2. The customer services portal will invoke the SCA Import using the with WebSphere Adapter for SAP Software for ALEREQ01 IDoc.

3. The adapter receives a request from the client application in the form of a Business Object. The adapter sends this data to SAP server in asynchronous form.

4. The SAP server after receiving the IDoc, tries to fetch the requested material Master data and responds with a transaction id. The one way transaction with the SAP server ends here.

5. Then SAP system later returns the requested data to the request sender asynchronously.

6. The SCA export configured with WebSphere Adapter for SAP Software listening for events pushed from the SAP system is active.

7. The adapter handles the response from SAP, converts it back to a Business Object format as required by the client application and sends it.

8. The Customer Services Portal returns the response to the user through email.

---

**Configuration prerequisites**

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or lipsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS , add the *.so libraries to the `<WPS_INSTALL>/lib` directory.
When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/sapjco3.jar` file to `WAS_SERVER_ONLY_server_region_classpath`.

The sapjco3.jar is required to run the New External Service wizard.

 `<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

Copy `CWYAP_SAPAdapterExt.jar` to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar` to `WAS_SERVER_ONLY_server_region_classpath`.

 `<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**.

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click on **Next**.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

**Figure: Select an Adapter screen**

**Note:** If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.

![Figure: Import a RAR file screen](image)
4. In the **Locate the Required Files and Libraries** screen, provide the location of the *sapjco3.jar* file and the *sapjco3.dll* or *libsapjco3.so* files.

5. Click **Next**.

6. In the **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.
Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

User name
Password
Host name
System number
SAP Client connection

Click Select to change the default Language code from English

Use the drop down option to change the default Code page from 1100.

Select ALE as the interface name
Click Next.

![WebSphere screen](image)

Figure: Select ALE as the interface

Selecting the Business Objects and services to be used with the adapter

In the Find objects in the Enterprise System screen, expand ALE under Discovered objects, click Discover IDoc From System and then click the button.
Enter Alereq01 (the name of the ALE in SAP system) in the Filter Properties for Discover IDoc From System screen.
Click **OK**.

Expand the **Discover IDoc From System** node and select ALEREQ01.
Click the button.

In the Configuration Properties screen, accept the default values by clicking on OK.
Figure: Setting configuration parameters for the IDoc ALEREQ01

ALEREQ01 has now been added to the list of Business Objects to be imported. Click Next.
In the **Specify Composite Properties** screen, enter *bodef* as the name of the Folder Name for the generated Business Object.
Click **Next**.

In the **Service Generation and Deployment Configuration** screen enter the deployment and connection information.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login into the SAP system.

Click Next

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select the Module radio button and click Next

In the New Module screen, type ALEOUT_ALEREQ01 in the Module Name field and click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module ALEOUT_ALEREQ01 by selecting **Add and Remove Projects**.

The project ALEOUT_ALEREQ01App will be listed under **Available projects**.

After adding the project, it should appear under the Configured projects.

Add the SCA module to the server by clicking Finish.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer’s Test Client.

Once the module is deployed to the Server, right click the module ALEOUT_ALEREQ01 from the Projects view and select Test > Test Module from the pop-up menu.

To test this tutorial, you use data from your SAP server. If you have not already done so, obtain actual values for the following data. If necessary, contact your SAP administrator to obtain the data.

Client
IdocNumber
SenderPort
PartnerNumberOfSender
ReceiverPort
PartnerNumberOfRecipient

Enter values in the Test Client as follows –
a) Enter the IDoc Control record data - Right-click SapAlereq01IDocBO and click **Add Element**

b) Type the following values in the associated fields:

- **ReceiverPort**
- **PartnerTypeOfSender**
- **NameOfBasicType**
- **PartnerNumberOfSender**
- **NameOfTableStructure**
- **Client**
- **LogicalMessageType**
- **PartnerNumberOfRecipient**
- **SenderPort**
- **IdocNumber**
- **PartnerTypeOfRecipient**

  c) Set the IDoc Data Record level property values.

- **Logicalmessagetype** - ALEREQ
- **Messagetype** - ALEREQ

Right-click SapAlereq01E2aleq1 and click Add Element

- **IncludingExcludingindicator** - E
- **RelationaloperatorEqNeGrLtGcLe** - LT
- **Lowerlimitforfieldcontents** - 0
- **Upperlimitforfieldcontents** - 1
Click the **Continue** button.

When the **Select Deployment** screen appears, select the WebSphere Process Server instance to which you added the project and click the **Finish** button.

If security is enabled, type in the username and password in the popup **User Login** screen that appears and click **OK**.
Chapter 11. Preparing to run through the AEP tutorial

Configuration prerequisites

Import the sample transports supplied along with the adapter into your SAP system.

The following is a list of the SAP R/3 version 4.7/ERP transport files necessary to support AEP module.

To ensure that all necessary tables are created before the data for those tables is added, the transports must be installed in the order listed. These files can be found in the directory <WID_INSTALLATION_DIRECTORY>\ResourceAdapters\SAP_7.0.0.0\transports.

<table>
<thead>
<tr>
<th>File name</th>
<th>Transport Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>47_Primary</td>
<td>VELK900031</td>
</tr>
<tr>
<td>47_Infrastructure</td>
<td>ERPK900137</td>
</tr>
</tbody>
</table>

The adapter requires the following libraries which are supplied by SAP.

Get the latest OS specific SAP JCO 3 libraries from the SAP Service Marketplace.

They should be copied to a folder on the system where the WebSphere Integration Developer’s EMD will be executed.

<table>
<thead>
<tr>
<th>SAP JCO files</th>
</tr>
</thead>
<tbody>
<tr>
<td>sapjco3.jar</td>
</tr>
<tr>
<td>sapjco3.dll</td>
</tr>
</tbody>
</table>
Replicas of the artifacts that you create when using the **New External Service wizard** are provided as sample files for your reference. Use these files to verify that the files you create with the **New External Service wizard** are correct.

Go to the **samples** folder and unzip **Tutorial_AEP.zip** into a directory of your choice (you may want to create a new directory).

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEPOutBoundSample/SAPOutboundInterface.import</td>
<td>Contains the SCA import for the resource adapter.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SAPOutboundInterface.wsdl</td>
<td>Service interface to invoke the resource adapter.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01.xsd</td>
<td>Business Object definition for the SapYxrv5b01.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01BG.xsd</td>
<td>Business Object definition for the Business Object graph.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv51000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv51000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv52000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv52000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv53000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv53000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv54000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv54000.</td>
</tr>
<tr>
<td>File Path</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv55000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv55000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv56000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv56000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv57000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv57000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv58000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv58000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv59000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv59000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv5a000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv5a000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv5b000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv5b000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01Z2xrv5c000.xsd</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv5c000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/DuplicateRecordFault.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/MatchesExceededLimitFault.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/MissingDataFault.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/ MultipleMatchingRecordsFault.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/ PrimaryKeyPairType.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/ RecordNotFoundFault.xsd</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>AEPOutBoundSample/ WBIFault.xsd</td>
<td>Fault Schema</td>
</tr>
</tbody>
</table>

**Triggering of events in the SAP System**
Chapter 12. Tutorial 8: AEP Interface outbound processing

This tutorial demonstrates how the WebSphere Adapter for SAP 7.0.0.0 uses the AEP interface to create, update, delete and retrieve a record from the Customer Master table. This tutorial explains how you can configure the adapter for outbound processing deploy and test the module for processing.

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `$ {WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath

The sapjco3.jar is required to run the New External Service wizard.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.
Copy CWYAP_SAPAdapterExt.jar to the $<WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add 
$\{WAS_INSTALL_ROOT\}/lib/ CWYAP_SAPAdapterExt.jar to 
WAS_SERVER_ONLY_server_region_classpath.

$<WPS_INSTALL> represents the WebSphere Process Server installation directory Configuring.

---

**the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: File -> New -> External Service.

1. Select Adapters > SAP from the **Select the Service Type of Registry** screen and 
click Next.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

*Figure: Select an Adapter screen*

**Note**: If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.

3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.
5. Click Next

6. In Select the Processing Direction screen, select the Outbound radio button and click Next.
Setting connection properties for the External Service wizard

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
- Click Select to change the default Language code from English
Use the drop down option to change the default Code page from 1100.

Select **Advanced event processing (AEP)** as the SAP Interface name. Click **Next**.

*Figure: Select AEP as the interface*
Selecting the Business Objects and services to be used with the adapter

In the Find Objects in the Enterprise System screen, expand AEP node under Discovered objects, select Discover IDoc From System, then click the button.

Figure: Find Objects in the Enterprise System
Enter Y* in the Filter Properties for Discover IDoc From System screen.

![Filter Properties for RFC](image)

*Figure: Filter Properties for RFC*

Click OK.

Expand the Discover IDoc From System node, select YXRV5B01 and click button.
In **Specify the Configuration Parameters for ‘YXRV5B01’**, click **Deselect All** button and then select only KUNNR (Customer Number 1) field. The **Customer Number 1** is the Primary Key field. Click OK.
Click **Next** on the Find objects in the Enterprise System screen.

**Generating Business Object definitions and related artifacts**

In the Specify Composite Properties screen, click the Add button and select Create from the Add Value pop up.

Enter the ABAP function module name and the relative folder for the generated Business Object as shown below.
Repeat the steps (a) and (b) above for Update, Delete and Retrieve operations. Give the ABAP function module name as mentioned below.

Update  - Y_XR_CUSTOMERMASTER_C2
Delete   - Y_XR_CUSTOMERMASTER_C3
Retrieve - Y_XR_CUSTOMERMASTER_C4

These are custom Function Modules in SAP system provided as samples by the adapter installer, which were called based on the operation.

Click Next.
On the **Service Generation and Deployment Configuration** screen, enter the connection and deployment information.
Figure: Service Generation and Deployment Configuration
**Note:** You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP.

Click **Next**.

In the Specify the Location Properties screen, click the **New** button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select **Module** radio button and click **Next**.

In the New Module screen, type **AEPOUT_YXRV5B01** in the Module Name field, and then click **Finish**.
Click **Finish** on Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module AEPOUT_YXRV5B01 by selecting Add and Remove Projects.

The project AEPOUT_YXRV5B01App will be listed under Available projects.
After adding the project, the added project should appear under the Configured projects. Add the SCA module to the server by clicking **Finish**.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer’s Test Client.

1. Once the module is deployed to the Server, right click on the module BAPI_CUST_GETDETAIL from the Projects view and select Test > Test Module from the pop-up menu.

2. Enter values in the Test Client as shown in the figure below

For Retrieve operation

**Operation:** Retrieve (retrieveSapYxrv5b01)

**Customernumber1:** 0000000802 (of BO SapYxrv5b01)

**Verb:** Create or Update or Delete
For the Create operation

Title: Mr

**Name:** Sample Test

**City:** Burlingame

Sortfield: IB

Customeraccountgroup: 0001

Characterfieldoflength11: EN

Transportationzonetoorfromwhichthegoodsaredelivered: 0000000001

Countrykey: US

ii. Add an element to SapYxrv5b01Z2xrv51000, and populate the following

Salesorganization: 0001

Distributionchannel: 01

Division: 01

Shippingconditions: 01

iii. Add an element to SapYxrv5b01Z2xrv54000, and populate the following

Nameofglobalcompanycode: 0001

Reconciliationaccountinggeneralledger: 120000

c) For Update operation

Customernumber1: 0000000815

**Title:** Mrs

3. Click the **Continue** button.

When the **Select Deployment** screen appears, select the WebSphere Process Server instance to which you added the project and click the **Finish** button.

4. If security is enabled, type in the username and password in the popup **User Login** screen that appears and click **OK.**
5. Check the output of the service in the test client

a. Retrieve

Return parameters
b. Create

---

**Return parameters**

---

```plaintext
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>createSap/AVB10/output</td>
<td>Sap/hv:AVB1</td>
<td>✓</td>
</tr>
<tr>
<td>verbo:verbo</td>
<td>Sap/hv:AVB1</td>
<td>✓</td>
</tr>
<tr>
<td>Customernumber1</td>
<td>Customernumber1:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Title</td>
<td>Title:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Translation</td>
<td>Translation:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Characterfieldlength1</td>
<td>Characterfieldlength1:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>RTTable</td>
<td>RTTable:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Authorizinggroup</td>
<td>Authorizinggroup:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Indentedkey</td>
<td>Indentedkey:verbo</td>
<td>✓</td>
</tr>
<tr>
<td>Characterfieldlength2</td>
<td>Characterfieldlength2:verbo</td>
<td>✓</td>
</tr>
</tbody>
</table>
```

---

c. Update – check the EIS.

6. Check that the data in the EIS matches the above output -

Login to the SAP GUI using the credentials for the given SAP server

Execute `/o/cwld/home_aep`, Click Management tab, Click Current Events button, enter Date of Event, click Execute
Clearing the sample content

If you have added a test record to the Customer Master table, clean up after this tutorial.
Chapter 13. Tutorial 9: AEP Interface Inbound processing

This tutorial demonstrates how the WebSphere Adapter for SAP 7.0.0.0 uses the AEP interface polling mechanism to retrieve events from the event table with P or Q status in the SAP system. These events will be processed by the adapter and sent to the configured end-points. After the events are retrieved and processed, they will be archived in the SAP system.

This tutorial explains how you can configure the adapter for inbound processing; deploy; and test the module for processing.

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath`
The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click **Next.**
Figure: Select the Service Type or Registry screen
2. Select the IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0) from the Select an Adapter screen and click Next.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node.
3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

5. Click **Next**.
6. In the **Select the Processing Direction** screen, select the **Inbound** radio button and click **Next**.

**Setting connection properties for the New External Service wizard**

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
SAP Client connection

Click Select to change the default Language code from English

Use the drop down option to change the default Code page from 1100.

Select **Advanced Event Processing (AEP)** as the SAP Interface name.

Click **Next**.

*Figure: Select AEP as the interface*
Selecting the Business Objects and services to be used with the adapter

In the **Find Objects in the Enterprise System** screen, expand **AEP** node under **Discovered objects**, select **Discover IDoc From System**, then click the button.

![Find Objects in the Enterprise System](image)

*Figure: Find Objects in the Enterprise System*
Enter Y* in the Filter Properties for Discover IDoc From System screen.

![Filter Properties for RFC](image)

*Figure: Filter Properties for RFC*

Click OK.

Expand the Discover IDoc From System node, select YXRV5B01 and click the button.
In Specify the Configuration Parameters for ‘YXRV5B01’, click Deselect All button and then select only KUNNR (Customer Number 1) field.

The Customer Number 1 is the Primary Key field. Enter the ABAP function module name as Y_XR_CUSTOMERMASTER_C1.

Click OK.
Figure: Setting configuration parameters for the IDoc YXRV5B01 selected

Click Next on the Find objects in the Enterprise System screen.

Generating Business Object definitions and related artifacts

In the Specify Composite Properties screen,
a) Select the IDoc YXRV5B01, click Add button under Service operations for selected IDoc and select Create from the Add Value pop up.

b) Enter the relative folder for the generated Business Object as bodefs.

Figure: Configuration properties for ‘YXRV5B01’

Click Next.

In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter an Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.

In the New Module screen, type AEPIN_YXRV5B01 in the Module Name field, and then click Finish.
Click **Finish** on the Specify the Location Properties screen.
Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the **AEPIN_YXRV5B01** SCA module, and double click the **Assembly Diagram**.

The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button of Java component from the **Palette**.
2. Click and drag the Java component to add the new component to the Assembly Diagram screen.

3. Add a Wire between the **SAPInboundInterface** and the Java component.

![AEP Inbound interface being wired to a target Component (end-point)](image)

4. In the Add Wire screen, click OK.

![Add Wire Confirmation Dialog](image)

5. Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
6. In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen is presented.

Edit the Java file if you want to write code to print trace and log messages or Data Object.

Ensure that the package com.ibm.j2ca.base.AdapterBOUtil is imported.

```java
/**
 * Method generated to support implementation of operation "emitCreateAfterImageSapYxrsv5b01" defined for WSDL port type
 * named "SAPInboundInterface".
 * The presence of common-sap.DataObject as the return type and/or as a parameter
 * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitCreateAfterImageSapYxrsv5b01(
    DataObject emitCreateAfterImageSapYxrsv5b01Input)
{
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSapYxrsv5b01Input));
    }
    catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Save the Java file and assembly diagram.

---

**Deploying the module in the test environment**

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer's Test Client.
Right click on your server node in the Server tab and add the module AEPIN_YXRV5B01 by selecting **Add and Remove Projects**.

The project AEPIN_YXRV5B01App will be listed under **Available projects**.

![Add and Remove Projects](image)

After adding the project, the added project should appear under the **Configured projects**. Add the SCA module to the server by clicking on **Finish**.
After the module is successfully deployed, you should see the below message on the Console:

Testing the assembled adapter application

Test the assembled adapter application by posting the events to Event Table. Adapter continuously polls for the events in this table.

Logon to the SAP system.

Events can be posted to event table by using one of the options.

a) Execute tcode `/o/cwld/home_aep, Management tab, Archived Events, click Execute, select an event, change the Event Status to P, save it, click continue, select the event, click
Resubmit button. Event will be posted to event table and gets processed by the adapter. This option is for testing purposes only.

b) Execute tcode XD01 and follow the below screen shots to create a customer record. After the customer is created, an event will be queued in the event table.

Click Continue.
ii. Save the customer record. See in the WebSphere Integration Developer that this event is picked up by the adapter and processed successfully.
iii. Execute tcode XD02 and follow the below screen shots to update a customer record. After the customer is updated, an event will be queued in the event table.

![Customer Change: Initial Screen](image)

iv. Click Continue
v. Save the customer record.

vi. See in the WebSphere Integration Developer that this event is picked up by the adapter and processed successfully.
Clearing the sample content

If you have added a test record to the Customer Master table, clean up after this tutorial.
Chapter 14. Troubleshooting AEP

1. **Symptom:** A ServiceRuntimeException exception is thrown at runtime:

    Caused by: com.ibm.websphere.sca.ServiceRuntimeException: Did not find method for native method 'emitCreateAfterImageSapYXRV5B01'
    ... 20 more

    javax.resource.ResourceException: Caught unexpected unchecked exception while delivering event to endpoint: $Proxy6@354e354e
    at com.ibm.j2ca.extension.eventmanagement.internal.EventSender.deliverEvent(EventSender.java:241)
    at com.ibm.j2ca.extension.eventmanagement.internal.EventSender.doSendEvent(EventSender.java:276)
    at com.ibm.j2ca.extension.eventmanagement.internal.EventSender.sendEvent(EventSender.java:191)
    at com.ibm.j2ca.extension.eventmanagement.internal.EventListSender.sendEvents(EventListSender.java:129)
    at com.ibm.j2ca.extension.eventmanagement.internal.EventListSender.run(EventListSender.java:99)
    at com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:419)
    at com.ibm.ejs.j2c.work.AsyncWorkProxy.run(AsyncWorkProxy.java:136)
    at com.ibm.ejs.j2c.work.AsyncWorkProxy.run(AsyncWorkProxy.java:85)
    at com.ibm.ejs.j2c.work.AsyncWorkProxy.run(AsyncWorkProxy.java:90)
    at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1469)

    Cause: This is usually caused by wrong BO name or Componenet1Impl which is not generated successfully.

    **Resolution:**

    Verify that the BO name is correct in the event table.

    Verify that the component1 is added in the Assembly Diagram and java class Componenet1Impl was generated.

    Run Menu Project, Clean., clean all projects.

    **Symptom:** Error received attempting to connect to SAP System with EMD

    **Resolution:**

    Verify that the connection parameters have been entered correctly
Chapter 15. **Tutorial 10: Sending data from the SAP system (inbound processing) and processing it using the ALE-passthrough interface with a Generic IDoc**

**Sending IDoc data From SAP – ALE Inbound Processing**

This tutorial demonstrates how to use New External Service to create a Generic Object for all IDocs, create an SCA module that uses the WebSphere Adapter (7.0) for SAP Software, deploy the module to the test environment of WebSphere Integration Developer (7.0).

In this tutorial, we use two IDoc types to test the Generic IDoc type – ALEREQ01 and MATMAS03.

**Scenario**

The scenario explains how ALE inbound processing can be used in real business integration case.

An integration developer wants to integrate a legacy system with a SAP ERP system.

Any changes in SAP system should be updated in the legacy system as well.

For example, a new customer record created in SAP should result in a new record legacy system.

The integration with the SAP system is handled by the WebSphere Adapter for SAP Software whereas the legacy system integration is handled by a proprietary connector.

In this case, the Integration developer has to use the ALE passthrough interface of the adapter in order to achieve the result.

The ALE passthrough interface can return the changes of SAP in the form of binary content which helps to configure SAP integration independent of a particular business item like customer or material, etc.
The user creates material information in SAP. The SAP user exit function triggers a matmas01 IDoc to the configured Receiving partner. In this case, the receiving partner is configured to SAP adapter.

The tutorial only cover the part of the scenario marked in red colored rectangular in the above picture.

1. The user creates Material information in the SAP system.

2. The user exit function in SAP triggers a Matmas01 IDoc to the adapter. The delivery is transactional; SAP creates a transaction ID which is used for Assured-event once delivery.

3. The adapter receives a request from the SAP system in the form of an IDoc object. The adapter converts the IDoc into a Generic Business Object. The Generic Business Object holds the IDoc content in a binary format.
4. The business process in WebSphere Process Server routes the Generic Business Object to subsequent components. In this case, the component is the one which can understand the IDoc in binary format for example, a WTX mapping.

5. Further in the business process you may want add more components which actual enrich the data received from adapter. This involves connecting other EIS or having static information. The enrichment component may or may not be useful depending on the legacy system requirements.

6. The enrichment component, delivers the enriched data to the legacy system.

The whole scenario is configured in asynchronous fashion. Any exception in the subsequent components should intimate the adapter component in a synchronous fashion so that the IDoc can be resubmitted from the SAP system.

---

**Configuration prerequisites**

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add ${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath.

The sapjco3.jar is required to run the New External Service wizard.
<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Application Server on z/OS, add 
${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar to 
WAS_SERVER_ONLY_server_region_classpath

---

<WPS_INSTALL> represents the WebSphere Process Server installation directory

Configuring the adapter for inbound processing

Run the New External Service wizard to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually Business Integration

Start the New External Service wizard by choosing: File -> New -> External Service

1. Select Adapters > SAP from the Select the Service Type of Registry screen and click Next.
Figure: Select the Service Type or Registry screen

The figure shows the Select the Service Type or Registry screen in WebSphere software, allowing users to choose the type of service to create or the registry to browse. The screen includes filters and available types such as Adapters, CICS, E-mail, FTP, JD Edwards, JDBC, Lotus Domino, Oracle, PeopleSoft, SAP, and Siebel. The description explains that this service allows users to access the SAP server using WebSphere® Adapter for SAP Software. It enables service-oriented integrated processes that can interact and exchange information with the SAP Server without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on the SAP EIS. During inbound processing, these services use the adapter to receive events from the SAP EIS.
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR File** and click **Next**.

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** screen, provide the location of the sapjco3.jar file and the sapjco3.dll or libsapjco3.so files.

![Figure: Locate the required files and Libraries screen](image)

5. Click **Next**.

6. In the **Select the Processing Direction**, select **Inbound** radio button, then click **Next**.
Setting connection properties for the New External Service wizard

You must prove the following information in the Discovery Configuration screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
- Click Select to change the default Language code from English
• Use the drop down option to change the default Code page from 1100.

• Select **ALE pass-through IDoc** as the SAP Interface name.

Click **Next**.

![Specify the Discovery Properties](image)

*Figure: Select ALE pass-through IDoc as the interface*

**Selecting the Business Objects and services to be used with the adapter**

Under **Find Objects in the Enterprise System**, expand the ALE node, select **Generic IDoc** and click the **>** button to add to the selected objects.
In the **Configuration Parameters** screen, choose the default values and Click **OK**.
Generic IDoc has now been added to the list of Business Objects to be imported.
Click **Next**

On the **Service Generation and Deployment Configuration** screen, enter the connection information. Click **Next**.
Figure: Service Generation and Deployment Configuration
**Note:** You can either enter the Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or enter the username and password used to login in to the SAP system.

Click Next.

In the **Specify the Location Properties** screen, click the **New** button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select **Module** radio button and click **Next**.

In the **New Business Integration Project** screen, type **ALEIN_PASSTHROUGH** in the **Module Name** field, and then click **Finish**.
Click Finish on the Specify the Location Properties screen.
Verify the results.

Figure: Artifacts created after the EMD run for ALE pass-through Inbound Module
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the ALEIN_PASSTHROUGH SCA module, and double click the Assembly Diagram. The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button of Java component from the Palette.

![Figure: ALE pass-through Inbound interface in the Assembly editor](image)

Click and drag the Java component to add the new component to the Assembly Diagram screen.

Add a Wire between the SAPInboundInterface and the Java component.
In the **Add Wire** screen, click **OK**.

![Add Wire Confirmation Dialog](image)

**Figure: Add Wire Confirmation Dialog**

Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.

![Generating Java Implementation](image)

**Figure: Creating Java implementation for the target Component.**

In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module ALEIN_PASSTHROUGH by selecting **Add and Remove Projects**.

The project ALEIN_PASSTHROUGHApp will be listed under **Available projects**.
After adding the project, the added project should appear under the **Configured projects**. Add the SCA module to the server by clicking on **Finish**.
Testing the assembled adapter application

Launch the SAP GUI.

Start the Transaction **WE19**.

Choose the radio button **Existing IDoc**.

Select an existing IDoc **MATMAS01** that you want to send

Set appropriate values in IDoc.

Select Standard Outbound Processing button

Select **Continue** in the pop-up box

This creates an ALE inbound event for the ALE inbound application deployed earlier.

The event should reach the Java end point, indicated by the output of **MATMAS01** Business Object on the console of WebSphere Integration Developer.
Chapter 16. Tutorial 11: Sending data to SAP (outbound processing) using the Queued RFC(qRFC) BAPI Interface

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses this interface to make calls to the SAP's native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector interface to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the <WPS_INSTALL>/bin directory (If WebSphere Integration Developer is installed generally the WebSphere Process Server instance is installed under <WID_INSTALL_DIR>/runtimes/bi_v7).

When working with WebSphere Process Server v7.0 on z/OS , add the *.so libraries to the <WPS_INSTALL>/lib directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy sapjco3.jar to the <WPS_INSTALL>/lib directory.

When working with WebSphere Process Server on z/OS, add ${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath.

The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.
When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Configuring the adapter for outbound processing

Run the New External Service wizard to generate Business Objects, Services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually Business Integration

Start the New External Service wizard by choosing: File -> New -> External Service

1. Select Adapters > SAP from the Select the Service Type of Registry screen and click Next.
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the Select an Adapter screen and click **Next**.
Note: If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node.

3. Specify the Connector Project name in the Import a RAR File screen and click Next.
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.
5. Click **Next**.

6. In **Select the Processing Direction** screen, select the **Outbound** radio button and click **Next**.
Setting connection properties for the External Service wizard

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
- SAP Client connection

Click Select to change the default Language code from English
Use the drop down option to change the default Code page from 1100.

Select BAPI as the SAP Interface name.

Click **Next**.

*Figure: Select BAPI as the interface*
Selecting the Business Objects and services to be used with the adapter

In the Find objects in the Enterprise System screen, click the RFC node. Then click the button.

*Figure: Object Discovery and Selection*
Enter **ZASYNCBAPI_1** (the name of the BAPI in SAP) in the **Filter Properties for ‘RFC’** screen.

![Figure: Filter Properties for RFC](image)

Click **OK**.

Expand the **RFC** node.
Select the **Z_ASYNCBAPI_1** from the metadata tree.

Click the **[ ]** button.

A popup will appear containing the Configuration properties for the **Z_ASYNCBAPI_1** object.

Check the **Use SAP filed names to generate attributes names checkbox** if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI.

Click **OK**.
Click Next

**Generating Business Object definitions and related artifacts**

In the Specify Composite Properties screen, check the Generate BAPI Business Objects within a wrapper checkbox.

Optionally, the Business Object folder name could be entered in which the Business Objects are created. Enter `bodefs` as the folder name.

Choose **Asynchronous Queued RFC** for SAP Remote Function Call (RFC) type.

Enter `Async_Bapi` for the Business object for **Service operations** field

Associate a Service Operation to the RFC Function `Z_ASYNCBAPI_1`.

To do this, Click the **Add** button under **Service Operations** and select **Create** and select the **RFC Function** for selection operation as `Z_ASYNCBAPI_1`. 
Choose the configured Queue name on the SAP Server for Select the queue name. TESTQUEUE is chosen here.

Click Next.

In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties.
**Figure: Service Generation and Deployment Configuration**

**Note:** You can either use the Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP.

Click Next.
In the **Specify the Location Properties** screen, click the **New** button next to the **Module** field to create a new module.

When the **New Business Integration Project** screen appears, select **Module** radio button and click **Next**.

![New Business Integration Project](image)

In the New Module screen, type **BAPI_qRFC_OUT** in the Module Name field, and then click **Finish**.
Click Finish on the Specify the Location Properties screen.
Verify the results.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer's Test Client.

Right click on your server node in the Server tab and add the module **BAPI_qRFC_OUT** by selecting **Add and Remove Projects**.

The project **BAPI_qRFC_OUTApp** will be listed under **Available projects**.
3. After adding the project, the added project should appear under the **Configured projects**. Add the SCA module to the server by clicking on **Finish**.
Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer test client.

Once the module is deployed to the Server, right click the module BAPI_qRFC_OUT from the Projects view and select Test > Test Module from the pop-up menu.

Enter values as shown in the following figure.

CustomerNumber1 0000000001

3. Click the Continue button.

When the Select Deployment screen appears, select the WebSphere Process Server to which you added the project and click the Finish button.
4. If security is enabled, type in the username and password in the popup User Login screen that appears and click OK.

5. Check the output of the service in the test client

6. Check the data in the EIS to ensure it matches expected values

Clearing the sample content

Nothing is required to clean up after this tutorial.
Chapter 17. Tutorial 12: Sending data from SAP system (INBOUND processing) using qRFC BAPI

Following sections explain inbound scenarios for the BAPI interface.

Configuration prerequisites

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath`

The sapjco3.jar is required to run the New External Service wizard.

 `<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

Copy `CWYAP_SAPAdapterExt.jar` to the `<WPS_INSTALL>/lib` directory.
When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar to
WAS_SERVER ONLY_server_region_classpath.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for inbound processing**

Run the **New External Service wizard** to generate business objects, services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**.

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click **Next**.
Select the Service Type or Registry

Select the type of service to create or registry to browse.

Filter: type filter text

Available Types:
- Adapters
  - CICS
  - Email
  - Flat File
  - FTP
  - IMS
  - iSeries
  - JDE J D Edwards
  - JDBC
  - Lotus Domino
  - Oracle
  - Peoplesoft
  - SAP
  - Siebel

Description:
Creates a service to access the SAP server using WebSphere® Adapter for SAP Software. You can create service-oriented integrated processes, which can interact and exchange information with the SAP Server, without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on the SAPEIS. During inbound processing, these services use the adapter to receive events from the SAP EIS.

Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

*Figure: Select an Adapter screen*

**Note:** If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify the Connector Project name in the **Import a RAR File** screen and click **Next**.
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

![Locate the Required files and Libraries screen](image)

*Figure: Locate the required files and Libraries screen*

5. Click **Next**.
6. In the **Select the Processing Direction** screen, select **Inbound** radio button, then click **Next**.

**Setting connection properties for the External Service wizard**

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select BAPI as the SAP Interface name.

Click **Next**.
Selecting the Business Objects and services to be used with the adapter

Under Find Objects in the Enterprise System, click RFC node. Then click the button.

*Figure: Object Discovery and Selection*
Enter Z_ASYNCBAPI_1 (the name of the BAPI in SAP) in the Filter Properties for ‘RFC’ screen.

Figure: Filter Properties for RFC

Click **OK**.

Expand the **RFC** node.
Select the ZASYNCBAPI_1 from the metadata tree.

Click the button.

A popup will appear containing the Configuration properties for the ZASYNCBAPI_1 object.

Check the Use SAP filed names to generate attributes names checkbox if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI.

Click OK.
Click Next.

**Generating Business Object definitions and related artifacts**

1. In the *Specify Composite Properties* screen, select *Create* as the operation, *bodef* as the folder name for the Business Objects generated and choose the RFC Function Call Type as *Asynchronous Transactional/Queued RFC*. 
2. Click **Next**.

3. In the **Service Generation and Deployment Configuration** screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter Authentication Alias already previously using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system.

4. Click Next.

5. In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

6. When the New Business Integration Project screen appears, select the Module radio button and click Next.

7. In the New Module screen, type BAPI_qRFC_IN in the Module Name field, and then click Finish.
8. Click Finish on the Specify the Location Properties screen.
9. Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the **BAPI_qRFC_IN** SCA module, and double click the Assembly Diagram. The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button of Java component from the **Palette**.

   ![Figure: BAPI qRFC Inbound interface in the Assembly editor](image)

Click and drag the Java component to add the new component to the **Assembly Diagram** screen.

Add a Wire between the **SAPInboundInterface** and the Java component.
In the Add Wire screen, click OK.

2. Right-click the new component and select Generate Implementation. This creates a Java component that will act as an endpoint.
In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.

```java
/**
 * Method generated to support implementation of operation "emitCreateAfterImageSapIAsyncAPI1Wrapper" defined for WSDL port type
 * named "SAPIboundInterface".
 * 
 * The presence of `com.ibm.xdo.DataObject` as the return type and/or as a parameter
 * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */

public void emitCreateAfterImageSapIAsyncAPI1Wrapper(
    DataObject emitCreateAfterImageSapIAsyncAPI1WrapperInput) {
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSapIAsyncAPI1WrapperInput));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Save the Java file

Save assembly diagram.
Deploying the module in the test environment

Test the assembled adapter application using the WebSphere Integration Developer’s Test Client.

Right click on the server and add the module **BAPI_qRFC_IN** by selecting Add and remove projects in WebSphere Integration Developer. You will see the project **BAPI_qRFC_INApp** listed under Available projects.

![Add and Remove Projects](image)

After adding the project, the added project should appear under the Configured projects. Add the SCA module to the server by clicking on **Finish**.
Testing the assembled adapter application

Launch the SAP GUI.

Define the destination in the sending system to enable qRFC communication. Use transaction SM59 to define the destination.

Register a RFC destination for qRFC. Use transaction SMQS. In the first screen, you will see a list of previously registered destinations. To register a new destination, choose REGISTRATION.
Enter the value for the RFC destination, MYRFCDESTINATION in the example.

Register a queue name. Invoke transaction SMQR. In the first screen, you will see a list of registered queue names.
Click Registration to add the new queue name.

At this point, your queue is setup and ready for use.
Invoke transaction SE37.

You need to have a new BAPI that initializes the queue and invokes the original BAPI and sends the result to the queue. So in EMD, you discover Z_ASYNCBAPI_1 but execute Z_QRFCBAPI_INBOUND1 from SE37 to send to the queue.
FUNCTION Z_QRFCBAPI_INBOUND.

**"Local Interface:
** IMPORTING
** " VALUE(CUSTOMERNO) TYPE BAPICUSTOMER_ID-CUSTOMER
** " VALUE(RFCDEST) TYPE RFCDES-RFCDEST
** " VALUE(QUEUENAME) TYPE TRFCOUT-QNAME
**

DATA: BAPI_CUSTOMERADDRESS LIKE BAPICUSTOMER_04,
      BAPI_CUSTOMERGENDETAIL LIKE BAPICUSTOMER_KNA1,
      BAPI_CUSTOMERCOMPANYDETAIL LIKE BAPICUSTOMER_05,
      BAPI_RETURN LIKE BAPIRET1,
      BAPI_CUSTOMERBANKDETAIL LIKE BAPICUSTOMER_02 OCCURS 6 WITH HEADER LINE.

CALL FUNCTION 'TRFC_QUEUE_INITIALIZE'.

CALL FUNCTION 'BAPI_CUSTOMER_GETDETAIL2'
  EXPORTING
  CUSTOMERNO = CUSTOMERNO
  COMPANYCODE =
  IMPORTING
  CUSTOMERADDRESS = BAPI_CUSTOMERADDRESS
  CUSTOMERGENDETAIL = BAPI_CUSTOMERGENDETAIL
  CUSTOMERCOMPANYDETAIL = BAPI_CUSTOMERCOMPANYDETAIL
  RETURN = BAPI_RETURN
  TABLES
  CUSTOMERBANKDETAIL = BAPI_CUSTOMERBANKDETAIL.

' Making RFEC BAPI call
CALL FUNCTION 'TRFC_SET_QUEUE_NAME'
  EXPORTING
    QNAME = QUEUENAME.

CALL FUNCTION 'Z_ASYNCBAPI_1'
  in background task
  destination RFCDEST
  EXPORTING
    CUSTOMERADDRESS = BAPI_CUSTOMERADDRESS
    CUSTOMERGENDETAIL = BAPI_CUSTOMERGENDETAIL
    CUSTOMERCOMPANYDETAIL = BAPI_CUSTOMERCOMPANYDETAIL
    RETURN = BAPI_RETURN
    TABLES
    CUSTOMERBANKDETAIL = BAPI_CUSTOMERBANKDETAIL.
Now, press **F8** to execute this BAPI

Here, you need to enter the queue name and the RFC destination

Invoke the Outbound queue monitor using transaction SMQ1

To view the specified queue, choose **Execute**

Double Click the queue name to see the LUWs.
Double click on **Execute LUW(F6)** to execute logical unit of work

The SAP system will send the LUW to the destination once it is alive.

In the console of WebSphere Integration Developer, you could see the BAPI Business Object printed (as we entered a print statement in the Component implementation above).

---

**Clearing the sample content**

Nothing is required to clean up after this tutorial.
Chapter 18. **Tutorial 13: Generating ALE Audit IDocs per packet (Inbound processing ALE Interface)**

Following sections explain inbound scenarios for the ALE interface where the ALE Audit are generated per packet rather than per IDOC.

**Configuration prerequisites**

**Note:** You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add `$ {WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath`.

The sapjco3.jar is required to run the New External Service wizard.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.
Copy `CWYAP_SAPAdapterExt.jar` to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add

$({WAS_INSTALL_ROOT})/lib/ CWYAP_SAPAdapterExt.jar to

WAS_SERVER_ONLY_server_region_classpath.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for inbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click **Next**.
Figure: Select the Service Type or Registry screen
1. Select the IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0) from the Select an Adapter screen and click Next.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the New External Service wizard earlier using the IBM WebSphere Adapter for SAP Software with transaction support in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the IBM WebSphere Adapter for SAP Software with transaction support node.
2. Specify a Connector Project name in the **Import a RAR File** screen and proceed by clicking on **Next**.

![Import a RAR file screen](image)

*Figure: Import a RAR file screen*
3. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

![Locate the required files and Libraries screen](image)

*Figure: Locate the required files and Libraries screen*

4. Click **Next**.
5. In the **Select the Processing Direction** screen, select **Inbound** radio button, then click **Next**.

---

**Setting connection properties for the New External Service wizard**

You must provide the following information in the **Discovery Configuration** screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select **ALE** as the SAP Interface name.

Click **Next**.

*Figure: Select ALE as the interface*
Selecting the Business Objects and services to be used with the adapter

Under Find Objects in the Enterprise System, expand the ALE node, click Discover IDoc From System.

![Object Discovery and Selection](image)

Click the button.

Enter `Alereq01` (the name of the ALE in SAP system) in the Filter Properties for ‘Discover IDoc From System’ screen.
Click **OK**.

Expand Discover IDoc From System node.

Select **ORDERS05** and click the button.
In the Configuration Parameters screen, choose the default values and click OK.
Figure: Setting configuration parameters for the ALE selected

ORDERS05 has now been added to the list of Business Objects to be imported.
Click **Next**.

**Generating Business Object definitions and related artifacts**

1. In the **Specify Composite Properties** screen, select the Service operation as **Create**.

2. Add `MessageType=ALEREQ; MessageCode=; MessageFunction=;` as IDoc Identifiers for the service operation by clicking on **Add** button.

3. Enter **bodefs** as the name of the relative folder for the generated Business Object.
4. Click **Next**

5. In the **Service Generation and Deployment Configuration** screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
NOTE: You can either enter Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system. Also enter the RFCProgramID (as shown in figure). This must have been already configured in the SAP system.

6. Click Advanced button, click ALE event status configuration and check the options ALE update status and Send ALEAUD per packet.

7. Click Next.

8. In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

9. When the New Business Integration Project screen appears, select Module radio button and click Next.
7. In the **New Module** screen, type **ALEIN_ALEAUDPERPACKET** in the **Module Name** field, and then click **Finish**.
10. Click **Finish** on the screen.

11. Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the ALEIN_ALEAUDPERPACKET SCA module, and double click the Assembly Diagram. The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button java component from the Palette.
2. Click and drag the java component to add the new component to the Assembly Diagram screen.

3. Add a Wire between the SAPInboundInterface and the Java component.

4. In the Add Wire screen, click OK.

Right-click on the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.

```java
public void emitCreateAfterImageSapOrders05(DataObject emitCreateAfterImageSapOrders05Input) {
    try {
        System.out.println(AdapterBOUtil.SerializeObject(emitCreateAfterImageSapOrders05Input));
    } catch (Exception e) {
        // To get or set attributes for DataObject emitCreateAfterImageSapOrders05Input, use the APIs as a
        // To get a string attribute in emitCreateAfterImageSapOrders05Input, use emitCreateAfterImage
        // To get a dataObject attribute in emitCreateAfterImageSapOrders05Input, use emitCreateAfterImage
    }
}
```

Save the Java file

Save assembly diagram.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module ALEIN_ALEAUDPERPACKET by selecting Add and Remove Projects.

The project ALEIN_ALEAUDPERPACKETApp will be listed under Available projects.

![Add and Remove Projects](image)

After adding the project, the added project should appear under the Configured projects. Add the SCA module to the server by clicking on Finish.
Testing the assembled adapter application

Launch the SAP GUI.

Start the Transaction WE20.

Open the appropriate partner profile and set the packet size for the message type configured during EMD.
In the Partner Profiles screen, double click on ALEREQ listed under Outbound parmtrs.

Change the **Packet Size** to 3 and select the radio button **Collect IDocs** and select the option **Collect IDocs**. Click **Save**.

Start the Transaction WE19.
Choose the radio button **Existing IDoc.**

Select an existing IDoc **ORDERS05** that you want to send.

Set appropriate values in IDoc.

Select Standard Outbound Processing button.

Change the No. of IDocs to be generated to any number. Select **Continue** in the pop-up box.

Note: If packet size is configured as 3 in SAP system and No. of IDocs to be generated is 6 then, it means that we are trying to send 2 packets. Hence 2 ALEAUD will be generated.

Start the transaction WE02. This opens the browser IDoc List. Give appropriate value for Creation On and click execute.

The next screen shows the Outbound IDocs and Inbound IDocs.

6 Outbound IDocs with packet size as 3 will result in 3 ALEAUD.

This creates an ALE inbound event for the ALE inbound application deployed earlier.
Chapter 19. Tutorial 14: ALE Audit support for Pass through IDocs (Generating ALEAUD per IDoc for ALE Pass through IDocs)

Configuration prerequisites

Note: You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the <WPS_INSTALL>/bin directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at <WID_INSTALL_DIR>/runtimes/bi_v7).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the <WPS_INSTALL>/lib directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the <WPS_INSTALL>/lib directory.

When working with WebSphere Process Server on z/OS, add ${WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath

The sapjco3.jar is required to run the New External Service wizard.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.
When working with WebSphere Application Server on z/OS, add
${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar to
WAS_SERVER_ONLY_server_region_classpath.

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for inbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration
to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service.**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and
   click **Next.**
Figure: Select the Service Type or Registry screen

Description:
Creates a service to access the SAP server using WebSphere® Adapter for SAP Software. You can create service-oriented integrated processes, which can interact and exchange information with the SAP Server, without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on the SAP EIS. During inbound processing, these services use the adapter to receive events from the SAP EIS.
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note:** If you have run the *New External Service* wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.

![Figure: Import a RAR file screen](image-url)
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

![Locate the required files and Libraries screen](image)

**Figure: Locate the required files and Libraries screen**

5. Click **Next**.
6. In the Select the Processing Direction screen, select Inbound radio button, then click Next button.

![Select the Processing Direction](image)

*Figure: Select the processing direction*

### Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

- User name
- Password
- Host name
- System number
SAP Client connection

Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select **ALE Pass-through IDoc** as the SAP Interface name.

   Click Next.

*Figure: Select ALE pass-through IDoc as the interface*
Selecting the Business Objects and services to be used with the adapter

Under **Find Objects in the Enterprise System**, expand the **ALE** node, select **Generic IDoc** and click the button to add to the selected objects.

![Image of Object Discovery and Selection](image)

*Figure: Object Discovery and Selection*

In the Configuration Parameters screen, choose the default values and click **OK**.
Figure: Configuration properties for ‘Generic IDoc’

Generic IDoc has now been added to the list of Business Objects to be imported.
Click **Next**.

On the **Service Generation and Deployment Configuration** screen, enter the connection information.
Figure: Service Generation and Deployment Configuration
NOTE: You can either enter Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or enter the username and password used to login in to the SAP.

Also enter the RFCProgramID (as shown in figure). This must have been already configured in the SAP system.

Click the Advanced tab and in ALE event status configuration select ALE update status.

Click Next.

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.
In the New Module screen, type **ALEIN_PASSTHOUGHAUDITPERIDOC** in the Module Name field, and then click **Finish**.
Verify results.

**Figure: Artifacts created after the EMD run for ALE pass-through Inbound Module**

**Generating Reference Bindings**

In the Business Integration Perspective of WebSphere Integration Developer, expand the ALEIN_PASTTHROUGHAUDITPERIDOC SCA module, and double click on the Assembly Diagram. The Assembly Diagram screen appears with the module’s Export component in view.

1. To create a new component, click the button of Java component from the Palette.
Click and drag the java component to add the new component to the Assembly Diagram screen. Add a Wire between the **SAPInboundInterface** and the Java component.

In the Add Wire screen, click OK.
Right-click on the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.

In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.
Save the Java file

Save assembly diagram.

Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer's Test Client.

Right click on your server node in the Server tab and add the module ALEIN_PASSTHOUGH_AUDITPERIDOC by selecting Add and Remove Projects.
Testing the assembled adapter application

Launch the SAP GUI.

Start the Transaction WE19.

Choose the radio button **Existing IDoc**

Select an existing IDoc **ALEREQ01** that you want to send

Set appropriate values in IDoc.

Select Standard Outbound Processing button

Select **Continue** in the pop-up box

Start the transaction WE02. This opens the browser IDoc List. Give appropriate value for Creation On and click execute.

The next screen shows the Outbound IDocs and Inbound IDocs.

An ALEAUD packet is generated for each outbound IDoc.

This creates an ALE inbound event for the ALE inbound application deployed earlier.

The event should reach the Java end point, indicated by the output of ALEREQ01 Business Object on the console of WebSphere Integration Developer.
Chapter 20. **Tutorial 15: Sending data from SAP (INBOUND processing) using tRFC BAPI**

Following sections explain inbound scenarios for the BAPI interface.

### Configuration prerequisites

**Note:** If you have previously configured prerequisites, skip this step and move to the next step.

After you create the connector project, you must add the required external dependencies into the project. SAP Java Connector interface is an external dependency that the adapter has for connecting to the SAP software application. The adapter uses this interface to make calls to the SAP native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector interface to the imported project. All external libraries and JAR files must first be copied to the appropriate locations on WebSphere Process Server:

Copy the dependency library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If WID is installed generally the WebSphere Process Server instance is installed under `<WID_INSTALL_DIR>/runtimes/bi_v7`).

(For z/OS users) add the *.so libraries to the `<WAS_INSTALL>/lib` directory.

(Windows users) Install the msvcp71.dll and msvcr71.dll files in the Windows system path.

You need the sapjco3.dll file to run the EMD wizard.

Copy sapjco3.jar to the `<WPS_INSTALL>/lib` directory.

(For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco3.jar` to `WAS_SERVER_ONLY_server_region_classpath`

You need the sapjco3.jar file to run the EMD wizard.

 `<WPS_INSTALL>` represents the WebSphere Process Server installation directory.
Copy CWYAP_SAPAdapterExt.jar to the <WPS_INSTALL>/lib directory.

(For z/OS users) add ${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar to WAS_SERVER_ONLY_server_region_classpath

<WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for inbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: File -> New -> External Service

1. Select **Adapters > SAP** from the Select the Service Type of Registry screen and click **Next**.
Select the Service Type or Registry screen

Create a service to access the SAP server using WebSphere® Adapter for SAP Software. You can create service-oriented integrated processes, which can interact and exchange information with the SAP Server, without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on the SAPEIS. During inbound processing, these services use the adapter to receive events from the SAP EIS.
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

**Note:** If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the **Import a RAR File** screen and click **Next**.
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

![Figure: Locate the required files and Libraries screen](image)

5. Click **Next**.
6. In the Select the Processing Direction screen, select Inbound radio button, then click Next button.

Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

- User name
- Password
- Host name
- System number
- SAP Client connection
Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select BAPI as the SAP Interface name.

Click Next.
Selecting the Business Objects and services to be used with the adapter

Under **Find Objects in the Enterprise System**, click on RFC node. Then click the button.

*Figure: Object Discovery and Selection*
Enter `Z_ASYNCBAPI_1` (the name of the BAPI in SAP) in the Filter Properties for ‘RFC’ screen.

![Filter Properties for RFC](image1)

**Figure: Filter Properties for RFC**

Click **OK**.

Expand the **RFC** node.

![Retrieved BAPIs’ based on search criteria](image2)

**Figure: Retrieved BAPIs’ based on search criteria**
Select the **Z_ASYNCBAPI_1** from the metadata tree.

Click the button.

A popup will appear containing the Configuration properties for the **Z_ASYNCBAPI_1** object.

Check the **Use SAP field names to generate attributes names checkbox** if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI.

Click **OK**.

![Figure: Setting configuration parameters for the BAPI 'Z_ASYNCBAPI_1'](image)

Click **Next**.
Generating Business Object definitions and related artifacts

In the Specify Composite Properties screen, select Create as the operation, bedefs as the folder name for the Business Objects generated and choose the RFC Function Call Type as Asynchronous Transactional/Queued RFC.

![Specify Composite Properties](image)

*Figure: Specify Composite Properties*

Click Next.

In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties.
Figure: Service Generation and Deployment Configuration
Note: You can either enter Authentication Alias previously created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP system. Also enter the RFCProgramID (as shown in figure). This must have been already configured in the SAP system.

Click Next

In the Specify the Location Properties screen, click the New button next to the Module field to create a new module.

When the New Business Integration Project screen appears, select Module radio button and click Next.

In the New Module screen, type BAPI_tRFC_IN in the Module Name field, and then click Finish.
Click Finish on the Specify the Location Properties screen.

Verify the results.
Generating Reference Bindings

In the Business Integration Perspective of WebSphere Integration Developer, expand the **BAPI_tRFC_IN** SCA module, and double click on the **Assembly Diagram**. The Assembly Diagram screen appears with the module's Export component in view.

1. To create a new component, click the button of Java component from the **Palette**.

2. Click and drag the Java component to add the new component to the Assembly Diagram screen.

3. Add a wire between the **SAPInboundInterface** and the Java component.
4. In the **Add Wire** screen, click **OK**.

5. Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
In the **Generate Implementation** screen, select the package in which the Java code will be created and click **OK**. A Java file in an editor screen appears.

Edit the Java file if you may wish to write code to print trace and log messages or Data Object.

Ensure that the package `com.ibm.j2ca.base.AdapterBOUtil` is imported.

```java
/**
   * Method generated to support implementation of operation "emitCreateAfterImageSapIAsyncSapi1Wrapper" defined for WSDL port type
   * named "SAPinboundInterface".
   *
   * The presence of com.ibm.wbo.DataObject as the return type and/or as a parameter
   * type conveys that it is a complex type. Please refer to the WSDL Definition for more information
   * on the type of input, output and fault(s).
   */

public void emitCreateAfterImageSapIAsyncSapi1Wrapper(DataObject emitCreateAfterImageSapIAsyncSapi1WrapperInput) {
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSapIAsyncSapi1WrapperInput));
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Save the Java file

Save assembly diagram.
Deploying the module in the test environment

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_qRFC_IN by selecting **Add and Remove Projects**.

The project BAPI_qRFC_INApp will be listed under **Available projects**.

After adding the project, the added project should appear under **Configured projects**. Add the SCA module to the server by clicking on **Finish**.
Testing the assembled adapter application

Launch the SAP GUI.

Start the transaction SE 3.

Enter the function module name as Z_ASYNCBAPI_INBOUND1 click execute (F8).
Give appropriate values to the fields CUSTOMERNO and RFCDEST. Click execute.

In the console of WebSphere Integration Developer, you could see the BAPI Business Object printed (as we entered a print statement in the Component implementation above).
Chapter 21. **Tutorial 16: Sending data to an SAP system (Outbound processing) using tRFC BAPI**

Following sections explain inbound scenarios for the BAPI interface.

**Configuration prerequisites**

*Note:* You do not have to perform this step if you have already done so for another scenario.

After you create the connector project, you must add the required external dependencies into the project. The SAP Java Connector (JCo) interface is an external dependency that the adapter requires in order to connect to SAP systems. The adapter uses SAP JCo to call SAP’s native interfaces.

Use WebSphere Integration Developer to add the SAP Java Connector library to the imported project. You must copy all external libraries and JAR files to the appropriate locations on the WebSphere Process Server:

Copy the native library (sapjco3.dll or libsapjco3.so files) to the `<WPS_INSTALL>/bin` directory (If the WebSphere Process Server v7.0 bundled with WebSphere Integration Developer v7.0 is used, it will be installed at `<WID_INSTALL_DIR>/runtimes/bi_v7`).

When working with WebSphere Process Server v7.0 on z/OS, add the *.so libraries to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Integration Developer v7.0 on Windows, ensure that msvcp71.dll and msvcr71.dll exist in the Windows system path.

The sapjco3.dll file is required to run the New External Service wizard.

Copy the sapjco3.jar file to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Process Server on z/OS, add

`$-{WAS_INSTALL_ROOT}/lib/the sapjco3.jar file to WAS_SERVER_ONLY_server_region_classpath`

The sapjco3.jar is required to run the New External Service wizard.

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.
Copy `CWYAP_SAPAdapterExt.jar` to the `<WPS_INSTALL>/lib` directory.

When working with WebSphere Application Server on z/OS, add `${WAS_INSTALL_ROOT}/lib/ CWYAP_SAPAdapterExt.jar` to WAS_SERVER_ONLY_server_region_classpath

`<WPS_INSTALL>` represents the WebSphere Process Server installation directory.

---

**Configuring the adapter for outbound processing**

Run the **New External Service wizard** to generate Business Objects, Services, and configuration to be used in this tutorial.

After opening WebSphere Integration Developer, the default perspective is usually **Business Integration**

Start the New External Service wizard by choosing: **File -> New -> External Service**

1. Select **Adapters > SAP** from the **Select the Service Type of Registry** screen and click **Next**.
Figure: Select the Service Type or Registry screen
2. Select the **IBM WebSphere Adapter for SAP Software with transaction support (IBM: 7.0.0.0)** from the **Select an Adapter** screen and click **Next**.

![Select an Adapter screen](image)

**Figure: Select an Adapter screen**

**Note**: If you have run the **New External Service** wizard earlier using the **IBM WebSphere Adapter for SAP Software with transaction support** in the current workspace, you can choose one from a list of configurations cached by WebSphere Integration Developer.

Select the correct configuration by expanding the **IBM WebSphere Adapter for SAP Software with transaction support** node.
3. Specify a Connector Project name in the Import a RAR File screen and proceed by clicking on **Next**.

*Figure: Import a RAR file screen*
4. In the **Locate the Required Files and Libraries** screen, provide the location of the `sapjco3.jar` file and the `sapjco3.dll` or `libsapjco3.so` files.

5. Click **Next**.
6. In the Select the Processing Direction screen, select Outbound radio button and click Next.

Setting connection properties for the New External Service wizard

You must provide the following information in the Discovery Configuration screen:

User name
Password
Host name
System number
SAP Client connection
Click Select to change the default Language code from English.

Use the drop down option to change the default Code page from 1100.

Select BAPI as the SAP Interface name.

Click **Next**.

*Figure: Select BAPI as the interface*
Selecting the Business Objects and services to be used with the adapter

In the **Find objects in the Enterprise System** screen, click RFC node. Then click the button.

![New External Service](#)

**Figure: Object Discovery and Selection**

Enter `bapi_bank_*` (the name of the BAPI in SAP) in the **Filter Properties for ‘RFC’** screen.
Click **OK**.

Expand the **RFC** node.

Select the **BAPI_BANK_CHANGE** from the metadata tree.

Click the **>** button.

A popup will appear containing the Configuration properties for the **BAPI_BANK_CHANGE** object.

Check the **Use SAP filed names to generate attributes names checkbox** if you want the Business Object attribute names to be generated using SAP field Names.

You can choose to create attributes in the Business Object for any optional parameter in the BAPI. Click **OK**.
Click OK.
Similarly add BAPI_BAND_CREATE, BAPI_BANK_GETDETAIL, BAPI_BANK_GETLIST and BAPI_BANL_SAVEREPLICA.
Click Next.
In the Service Generation and Deployment Configuration screen enter the connection properties and deployment properties.
Note: You can either enter Authentication Alias already created using the Administrative Console of the WebSphere Process Server or simply enter the username and password used to login in to the SAP.

Click Next.
In the **Specify the Location Properties** screen, click the **New** button next to the Module field to create a new module.

When the **New Business Integration Project** screen appears, select **Module** radio button and click **Next**.

In the New Module screen, type **BAPI_qRFC_OUT** in the Module Name field, and then click **Finish**.
Click Finish on the Specify the Location Properties screen.
Click **Finish**.

Verify the Results.
The WSDL file generated during the **New External Service** wizard run shows that more than four BAPIs can be selected at the same time.
**Deploying the module in the test environment**

After completing the New External Service wizard, an SCA module gets generated with EIS Import or Export options. This module must be installed in the WebSphere Integration Developer’s Test Client.

Right click on your server node in the Server tab and add the module BAPI_qRFC_IN by selecting **Add and Remove Projects**.

The project BAPI_qRFC_INApp will be listed under **Available projects.**
Click **Finish**.
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR
PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication.

IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites.

The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation Department
2Z4A/SOM1 294 Route 100
Somers, NY 10589-0100 U.S.A.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include

the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute

...
these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning:

Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

**Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Screens are trademarks of Microsoft Corporation in the United States, other countries, or both.
Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.*

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org).