IBM WebSphere Adapter for SAP Software
6.2.0.0
Quick Start Scenario

September 18, 2008
• This edition applies to version 6, release 2, modification 0 of IBM WebSphere Adapter for SAP Applications and to all subsequent releases and modifications until otherwise indicated in new editions.

• © Copyright International Business Machines Corporation 2008. US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Chapter 1. Introduction

Chapter 2. Hardware and software prerequisites

Chapter 3. Installing the adapter

Deployment prerequisites

Configuring the authentication alias on the process server

Chapter 4. Tutorial 1: Sending data to SAP (outbound processing) using simple BAPI Interface

Configuration prerequisites

Configuring the adapter for outbound processing

Setting connection properties for the external service wizard

Selecting the business objects and services to be used with the adapter

Generating business object definitions and related artifacts

Deploying the module to the test environment

Testing the assembled adapter application

Clearing the sample content

Chapter 5. Tutorial 2: Sending data to SAP (outbound processing) using BAPI Work Unit Interface

Configuration prerequisites

Configuring the adapter for outbound processing

Setting connection properties for the external service wizard

Selecting the business objects and services to be used with the adapter

Generating business object definitions and related artifacts

Deploying the module to the test environment

Testing the assembled adapter application

Clearing the sample content

Chapter 6. Tutorial 3: Sending data to SAP (outbound processing) using BAPI ResultSet Interface

Configuration prerequisites

Configuring the adapter for outbound processing

Setting connection properties for the external service wizard

Selecting the business objects and services to be used with the adapter

Generating business object definitions and related artifacts
Chapter 7. Tutorial 4: Sending data from SAP (INBOUND processing) using BAPI

Configuration prerequisites
Configuring the adapter for inbound processing
  Setting connection properties for the external service wizard
  Selecting the business objects and services to be used with the adapter
  Generating business object definitions and related artifacts

Deploying the module to the test environment
Testing the assembled adapter application
Clearing the sample content

Chapter 8. Tutorial 5: Sending Structured Query to SAP – Query Outbound Processing

Configuration prerequisites
Configuring the adapter for outbound processing
  Setting connection properties for the external service wizard
  Selecting the business objects and services to be used with the adapter
  Generating business object definitions and related artifacts

Deploying the module to the test environment
Testing the assembled adapter application
Clearing the sample content

Chapter 9. Tutorial 6: Sending data from the SAP (inbound processing) using simple ALE Interface

Sending IDoc data From SAP – ALE Inbound Processing
Configuration prerequisites
Configuring the adapter for inbound processing
  Setting connection properties for the external service wizard
  Selecting the business objects and services to be used with the adapter

1.5 Deploying the module to the test environment
Testing the assembled adapter application

Chapter 10. Tutorial 7: Sending data to SAP (outbound processing) using simple ALE Interface
Chapter 16. Tutorial 11: Sending data to SAP (outbound processing) using Queued RFC(qRFC) BAPI Interface

Chapter 17. Tutorial 12: Sending data from SAP (INBOUND processing) using qRFC BAPI
Chapter 1. Introduction

- This document demonstrates how the WebSphere Adapter for SAP Software could be used for providing business integration connectivity.

- The BAPI outbound sample application demonstrates how the WebSphere Adapter for SAP Software can be used to create business objects based on various BAPI customer-related function calls. The sample shows how to configure the adapter as an SCA component and, once deployed, how to invoke it to create a Customer object in SAP by invoking the BAPI.

- The ALE Inbound sample scenario 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 in order 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 some complex scenarios related to ALE and BAPI interfaces.
Chapter 2. Hardware and software prerequisites

- Install the prerequisite software listed below:

- For a complete hardware software requirements please visit –

- IBM WebSphere Adapter for SAP Software HW/SW requirements
Chapter 3. Installing the adapter

The adapters are installed with WID and are located in the <WID_installation>/ResourceAdapters/SAP_6.2.0.0/deploy directory.

Note the following:

- The adapter RAR file contains both the adapter and the external service wizard files.

- Use CWYAP_SAPAdapter.rar file if you want the adapter to perform ALE, BAPI, SQI & AEP processing without local transaction support. In this case, the application provides the local transaction support.

- Use CWYAP_SAPAdapter_Tx.rar file if you want the container (WebSphere Process Server) to control local transaction support for BAPI transaction and ALE outbound transaction processing. In this case, the adapter participates in the local transaction started by the container.

- 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 6.2 (WebSphere Integration Developer)

- WebSphere Process Server 6.2 administrative console

- For WebSphere Process Server installation instructions, see the WebSphere Process Server documentation.

- After you ensure that above steps are executed, make sure you know the following information for accessing the SAP application:

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

• Get SAP JCO library, you can get sap jco from [https://websmp101.sap-ag.de/](https://websmp101.sap-ag.de/). Please get the userId/Password for this URL from BASIS person.

---

**Configuring the authentication alias on the process server**

• Before you install the application, you must create an authentication alias for use with your SAP instance. Once an authentication alias has been created, other SAP application project modules may use it as well.

• Starting Websphere Process Server Administrative console:

• Select “Business Integration” prospective in WID. Under “Server” tab right click on “Websphere Process Server v6.2” and select “Run Administrative Console” as shown in screenshot below.
In the WebSphere Process Server Administrative Console click Security > Secure administration, applications, and infrastructure.

Then select Authentication > Java Authentication and Authorization Service > J2C authentication data.

If an alias named “SAP_Auth_Alias” does not already exist, create it now.

Click “New”. The General properties window appears.
• In the “Alias” field, specify “SAP_Auth_Alias” (Remember this authentication alias as its required while generating inbound\outbound services for Adapter).

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

• Click OK

• Click Save.
Troubleshooting: While running the scenario with a new RAR file for first time, in the Required Files and Libraries it may ask for one more dll file (if not present in the path). Its path has to be specified.

![Required Files and Libraries](image)

- SAP sapico.jar file:
- System library file (libr32.dll or libr32.so):
- System library file (sapjco32.dll or libsapjco32.so):
- Library file (msvcr71.dll):

When deploying the application to a stand-alone server, these files must be configured on the server. More...
Chapter 4. Tutorial 1: Sending data to SAP (outbound processing) using simple BAPI Interface

Following sections explain outbound scenarios for the BAPI interface.

Configuration prerequisites

- Note: If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.o, librfc32.o files) to the <WPS_INSTALL>\bin directory (If WID is installed generally the WPS instance is installed under <WID_INSTALL_DIR>\runtimes\bi_v6 ).

- (For z/OS users) add the *.so libraries to the <WAS_INSTALL>/lib directory.

- (Windows users) Install the msvcp71.dll and msver71.dll files in the Windows system path.
• 3. Copy sapjco.jar to the `<WPS_INSTALL>\lib` directory.

• (For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco.jar` to WAS_SERVER_ONLY_server_region_classpath

• Also you need this jar while running EMD wizard.

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

---

**Configuring the adapter for outbound processing**

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File -> New -> External Service
3. Select SAP Adapter then click Next

4. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node. Further select the Connection that matches the
SAP Server and connection credentials. Then click next.

5. In the Connector Project Settings Screen, provide the path for “sapjco.jar”.
   Also provide the path for the two System Library files.
• 6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Outbound” radio button, then click “Next >” button.

To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI”. Then click next.
Select BAPI as the interface

Selecting the business objects and services to be used with the adapter

1. Under “Object Discovery and Selection”, click on RFC node. Then click on the “go” button.
- Figure: Object Discovery and Selection

- 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 Discover by Name window.
- Figure: Filter Properties for RFC
- Click “OK”.
- Expand “RFC” node.
Figure: Retrieved BAPIs’ based on search criteria

Select the “BAPI_CUSTOMER_GETDETAIL” from the metadata tree.

Click on the “button. Clicking this should bring up the Configuration Parameters for the “BAPI_CUSTOMER_GETDETAIL”. This screen has 2 fields. The first field “Use SAP field names to generate attributes names” may be checked if attribute names need to be generated using SAP field Names. Second element helps to select any optional parameter that the user might be interested in adding to the business objects. Click “OK”
Figure: Setting configuration parameters for the BAPI selected

7. Click “Next”.

Generating business object definitions and related artifacts

Follow these steps to generate the business object definitions.

In the “Configure Composite Properties”, enter the business Object name for the service description. Next step is to associate a operation with the Business Object. To do that, select a service operation and then select Business Object in the next drop-down. Also the user can enter the name of the folder where the business objects are created. Click “OK” button to go back to “Object Discovery and Selection” screen.
• Figure: Configure Composite Properties

• 2. Click “Next”.
3. On the Service Generation and Deployment Configuration screen enter the connection information. Click next. Please Note that J2C authentication name should be “<server node name>/ SAP_Auth_Alias

- Figure: Service Generation and Deployment Configuration
• Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply enter the username and password used to login in to the SAP.

• 4. Enter the username and password and click next.

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

• 6. If the New Integration Project screen appears, select Create a module project, then click Next

• 7. In the New Module screen, type BAPI_CUST_GETDETAIL in the Module Name field, and then click Finish.
8. Click Finish.

9. Verify the results.
• Figure: Artifacts created after running the Simple BAPI Module

Deploying the module to the test environment

• The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

• 1. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select Add and remove projects. You should see the project listed under “Available projects”.
2. 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.

- Select the BAPI_CUST_GETDETAIL, right click and select Test > Test Module from the pop-up menu.

- Enter values as shown in the following figure.

- CustomerToBeRequired 0000000001
- Distribution Channel 01
- Sales Organization 0001
- Division 01
3. Click on the Continue button. When the Select Deployment window appears, select the WPS server to which you added the project and click on the Finish button.

4. Click on OK

Check the output of the service, and check the data in the EIS to ensure it matches expected values.
Clearing the sample content

- Return the data to its original state.

  Nothing is required to clean up after this tutorial.
Chapter 5. Tutorial 2: Sending data to SAP (outbound processing) using BAPI Work Unit Interface

- Following sections explain outbound scenarios for the BAPI work Unit interface.

Configuration prerequisites

- Note: If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.o, librfc32.o files) to the <WPS_INSTALL>\bin directory (If WID is installed generally the WPS instance is installed under <WID_INSTALL_DIR>\runtimes\bi_v6).

- (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.
• Copy sapjco.jar to the `<WPS_INSTALL>/lib` directory.

• (For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco.jar` to WAS_SERVER_ONLY_server_region_classpath

• Also you need this jar while running EMD wizard.

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

---

**Configuring the adapter for outbound processing**

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File -> New -> External Service
3. Select SAP Adapter then click Next

4. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node. Further select the Connection that matches the
SAP Server and connection credentials. Then click next.

5. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files. `Librfc32.dll/librfccm.so`
and sapjcorfc.dll/libsapjcorfc.so files.

6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Outbound” radio button, then click “Next >” button.

- To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI Work Unit”. Then click next.
Selecting the business objects and services to be used with the adapter

- Under “Object Discovery and Selection”, click on RFC node. Then click on the “+” button.
2. Enter BAPI_CUSTOMER* (the name of the BAPI in SAP plus an asterisk as a wild card character) in the Filter Properties for Discover by Name window.
- Click “OK”.
- Expand “RFC” node.

Figure: Filter Properties for Discover By Name
5. Select the BAPI_CUSTOMER_CREATEPWREG and BAPI_CUSTOMER_DELETEPWREG.

6. Click on the “قدرة” button. Clicking this should bring up the Configuration Parameters for both the BAPIs. Select any of the BAPIs to configure it.
• Figure: Setting configuration parameters for the BAPI selected

• The field “Use SAP field names to generate attributes names” may be checked if attribute names need to be generated using SAP field Names. Click “ok” button to go back “Object Discovery and selection” screen

• Click “Next”.

Generating business object definitions and related artifacts

• Follow these steps to generate the business object definitions.

• In the “configure Objects Screen”, enter the business Object name for the service description. Next step is to associate a operation with the Business Object. To do that, select a operation (by clicking ADD button) and then select Business Object in the next drop –down. Also the user can enter the name of the folder where the business objects are created.
- Select create and click "OK". Now the operation is selected.
Now we need to associate a Business Object to this operation. Select the BAPI “BAPI_CUSTOMER_CREATEPWREG”. Click on the “ADD” button to do that.
1. To add one more service operation, click on “ADD” button next to service operation. Now pop-up window will display only the remaining service operations. Select the “Delete” operation.
• Now click “ADD” located next to Business Objects. Select “BAPI_CUSTOMER_DELETEPWREG”. Now you have selected 2 combinations of Service operations and Business Objects.

• 3. To verify the selections. Just click on any service operation selected and its corresponding business object should be displayed automatically in the text box below it.
4. Click “Next”.
1. On the Service Generation and Deployment Configuration screen enter the connection information. Click next.

- Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply enter the username and password used to login in to the SAP.
6. Enter the username and click next.

7. In the Service Location Properties screen, click the New button next to the Module field to create a new module. By default we see name for module. WID retains this name from last time around when we created a module.

Figure 27
8. If the New Integration Project screen appears, select **Create a module project**, then click **Next**

9. In the Module screen, type **BAPI_CUST** in the Module Name field, then click **Finish**.
• Figure 29

• 10. Click Finish.

• 11. Verify the results.
Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

- Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**. Select BAPI_CUSTApp.
• Once selected then click on “ADD” button. It should appear under the Configured Projects.
Add the SCA module to the server.

**Testing the assembled adapter application**

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

- In “Business Integration” view, right click on the “BAPI_CUSTApp” project

- Populate values for input business objects.

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

3. Click on the button. When the Select Deployment window appears, select the WPS server to which you added the project and click on the Finish button.

4. Click on OK

Check the output of the service, and check the data in the EIS to ensure it matches expected values.
Clearing the sample content

- Return the data to its original state.

Nothing is required to clean up after this tutorial.
Chapter 6. Tutorial 3: Sending data to SAP (outbound processing) using BAPI ResultSet Interface

- Following sections explain outbound scenarios for the BAPI ResultSet interface.

**Configuration prerequisites**

- Note: If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.so, librfc32.so files) to the `<WPS_INSTALL>\bin` directory (If WID is installed generally the WPS instance is installed under `<WID_INSTALL_DIR>\runtimes\bi_v6`).

  - (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.

  - Copy sapjco.jar to the `<WPS_INSTALL>\lib` directory.
• (For z/OS users) add \${WAS_INSTALL_ROOT}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath

• Also you need this jar while running EMD wizard.

• Where <WPS_INSTALL> represents the WebSphere Process Server installation directory.

-----------------------------

**Configuring the adapter for outbound processing**

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File -> New -> External Service
Select SAP Adapter then click Next

1. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node. Further select the Connection that matches the SAP Server and connection credentials. Then
2. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files. Librfc32.dll/librfcem.so
6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Outbound” radio button, then click “Next >” button.

To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI ResultSet”. Then click next.
Selecting the business objects and services to be used with the adapter

- Under “Object Discovery and Selection”, click on RFC node. Then click on the “<” button.
Enter bapi_cust_get* (the name of the BAPI in SAP plus an asterisk as a wildcard character) in the Filter Properties for Discover by Name window.
Figure: Filter Properties for Discover By Name

- Click “OK”.
- Expand “RFC” node.
- Figure: Selected BAPI added to the Objects to be imported

- Select the BAPI_CUSOMTER_GETDETAIL and BAPI_CUSTOMER_GETLIST.

- Click on the “>” button. Clicking this should bring up the Configuration Parameters for the both the BAPIs. Select any of the BAPIs to configure it.
Figure: Setting configuration parameters for the BAPI selected

This screen has 2 fields. The first field “Use SAP field names to generate attributes names” may be checked if attribute names need to be generated using SAP field Names. Second element helps to select any optional parameter that the user might be interested in adding to the business objects. ” click “Ok” button to go back to “Object Discovery and Selection” screen

Click “Next”.

Generating business object definitions and related artifacts

Follow these steps to generate the business object definitions.
• In the “Configure Composite Properties”, Business Name map the field in “Query BAPI” to a field in “Result BAPI”. To do that, select a BAPI as Query BAPI and the other BAPI is automatically selected as Result BAPI.

• Then click on “ADD” Button. This brings up a pop-up menu. It lists both the BAPIs required for the result set.
Click the “select” button corresponding to SapBapiCustomergetdetail and select \texttt{SapBapiCustomergetdetail/CustomerNumberToBeRequired} and from SapBapiCustomerGetlist select \\
\texttt{SapBapiCustomerGetlist/SapAddressdata/CustomerNumber1} \\
Click “Finish” button to go back to “Configure Composite Properties. Enter folder name and Click Next.
Configure Composite Properties
Specify properties that apply to all selected objects.

Enter the name of the business object: Customer
Query BAPI: BAPI_CUSTOMER_GETLIST
Result BAPI: BAPI_CUSTOMER_GETDETAIL

Specify one or more foreign key relationships between the query BAPI and the result BAPI:

<table>
<thead>
<tr>
<th>SapBapiCustomerGet</th>
<th>SapBapiCustomerGetlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>SapBapiCustomerGet</td>
<td>SapBapiCustomerGetlist/SapAddressdata/CustomerNumber1</td>
</tr>
</tbody>
</table>

Specify the relative folder for generated business object:
Folder: bodefs

Generate a business graph for each business object
Ignore errors in BAPI Return object

Figure: Configure Objects- Adding a Service Operation
• On the Service Generation and Deployment Configuration screen enter the connection information. Click next.

• Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply enter the username and password used to login in to the SAP.

• Enter the username and password and click next.
Service Generation and Deployment Configuration

Warning: Sensitive values, such as passwords, should not be saved.

Service operations:
If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties:
- Specify a Java Authentication and Authorization Services (JAAS) alias security credential.
- J2C Authentication Data Entry: widNode/SAP_AUTH_ALIAS
- Deploy connector project: With module for use by single application
- Specify the settings used to connect to SAP Software at runtime:
  - Connection properties: Specify connection properties

Connection properties:
- SAP system connection information
  - Host name: saperp05.sv.ltm.com
  - System number: 00
  - Client: 100
  - Language code: EN (English)
  - Code page: 1100

The user name and password will not be encrypted and will be stored as plain text.
- User name: zkhan
- Password: ********

Advanced >>
- In the Service Location Properties screen, click the **New** button next to the Module field to create a new module. By default we see name for module. WID retains this name from last time around when we created a module.

- Figure - Publishing Properties

- If New…” button was clicked the New Integration Project screen appears, select **Create a module project**, then click **Next**
In the New Module screen, type **BAPI_CUST3** in the Module Name field.
Business integration modules can be deployed and run on WebSphere Process Server. They can contain many types of components, such as business processes, assembled together for the purpose of business integration.

- Figure 29

- Click Finish.

- Verify the results.
Figure: Artifacts generated after BAPI ResultSet interface is run

### Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

- Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**.

- Add the module and click finish.
Add the SCA module to the server.

Testing the assembled adapter application

- Test the assembled adapter application using the WebSphere Integration Developer integration test client.
- Right click and select Test > Test Module from the pop-up menu.
- Populate values for input business objects.
- Right click on SapBapiGetDetail and select “AddElements” and enter 1. Enter following values.
  - CustomerToBeRequired= 0000000001
• DistributionChannel=01
• Division=01
• SalesOrganization=0001
• Under QueryBO set the following data
• MaximumNumberOfCustomers= 3

• In QueryBO we have another element with name “SapIdRange”
• Right click on SapIdRange and select “AddElements” and enter 1.
• InclusionExclusionCriterionSignForRangeTables= E
• SelectionOperatorOptionForRangeTables= EQ
• CustomerNumber1=10
• CustomerNumber2217378=25
• Under the QueryBO we also have SapAddressData.
• Right click on SapAddressData and select “AddElements” and enter 1.
• CustomerNumber=0000000001
4. Click on the Continue button. When the Select Deployment window appears, select the WPS server to which you added the project and click on the Finish button.

5. Click OK.

Check the output of the service, and check the data in the EIS to ensure it matches expected values.

Clearing the sample content
• Return the data to its original state.

Nothing is required to clean up after this tutorial.
Chapter 7. Tutorial 4: Sending data from SAP (INBOUND processing) using BAPI

- Following sections explain inbound scenarios for the BAPI interface.

Configuration prerequisites

- Note: If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.o, librfc32.o files) to the <WPS_INSTALL>\bin directory (If WID is installed generally the WPS instance is installed under <WID_INSTALL_DIR>\runtimes\bi_v6).

- (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.
• Copy sapjco.jar to the `<WPS_INSTALL>\lib` directory.

• (For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco.jar` to WAS_SERVER_ONLY_server_region_classpath

• Also you need this jar while running EMD wizard.

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

---

**Configuring the adapter for inbound processing**

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File-> New -> External Service
1. Select SAP Adapter then click Next

2. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node. Further select the Connection that matches the
SAP Server and connection credentials. Then click next.

3. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files, Librfc32.dll/librfccm.so and
sapjcorfc.dll/libsapjcorfc.so files.

- 6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Inbound” radio button, then click “Next >” button.

- To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI”. Then click next.
Selecting the business objects and services to be used with the adapter

1. Under “Object Discovery and Selection”, click on RFC node. Then click on the “” button.
Figure: Object Discovery and Selection

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 Discover by Name window.
3. Click “OK”.
4. Expand “RFC” node.
1. Select the “BAPI_CUSTOMER_GETDETAIL” from metadata tree.

Click on the “button. Clicking this should bring up the Configuration Parameters for the “BAPI_CUSTOMER_GETDETAIL”. This screen has 2 fields. The first field “Use SAP field names to generate attribute Names” may be checked if attribute names need to be generated using SAP field Names. Second element helps to select any optional parameter that the user might be interested in adding to the business objects. Click “OK” button to go back to “Object Discovery and Selection” screen.
Generating business object definitions and related artifacts

- Follow these steps to generate the business object definitions.

- 1. In the “configure Composite properties” Screen, associate the RFC-enabled function name with an end-point operation. Enter the Business Object namespace. Also the user can enter the name of the folder where the business objects are created.
Figure 8: configure Composite properties

2. Click “Next”.
3. On the Service Generation and Deployment Configuration screen enter the connection information. Click next.

![Service Generation and Deployment Configuration](image)

- **Figure: publishing Object Configuration Properties**
• Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply 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.

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

• If the New Integration Project screen appears, select **Create a module project**, then click **Next**

5. In the New Module screen, type BAPI_CUST_GETDETAIL_IN in the Module Name field, and then click Finish.
6. Click Finish.

7. Verify the results.
7.2.4 Generating Reference Bindings

- In the Business Integration Perspective of WebSphere Integration Developer, right-click the “BAPI_CUST_GETDETAIL_IN” SCA module, and select Open Assembly Diagram. The Assembly Diagram window appears with the module’s Export component in view.

1. To create a new component, click the button of java component from the “Palette”.

- BAPI Inbound – Figure 13: BAPI Inbound interface in the Assembly editor

- Click the palette to add the new component to the Assembly Diagram window.
Click and drag the java component to the new component. This draws a wire from the Export component to the new component and displays the Add Wire window.

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

5. In the Add Wire window, click OK.

ALE Inbound – Figure 15: Confirmation to create an export for the target Java Component (wiring)

4. Right-click the new component and select Generate Implementation. This creates a Java component that will act as an endpoint.
Figure: Creating Java implementation for the target Component.

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

Edit the Java file. For example, you may wish to write code to print trace and log messages or Data Object.
• Save the Java file

• Save assembly diagram.

---

**Deploying the module to the test environment**

• The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

• Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**. The project will be seen under “Available projects”. 
Figure: Available projects

Click on the Add button and it should be transferred into the configured projects.
Figure: Configured projects

- Add the SCA module to the server. Click finish to complete it

Testing the assembled adapter application

- Launch the SAP GUI. Start the Transaction SE37. Enter the BAPI “BAPI_CUSTOMER_GETDETAIL”. Click on Execute button.

- In the “Test Function Module” screen, Enter RFC Target Sys as the RFCProgramID that has already been configured in SAP. Enter other data as shown in the figure.
3. Execute the BAPI.

Clearing the sample content

- Return the data to its original state.

Nothing is required to clean up after this tutorial.
Chapter 8. **Tutorial 5 Sending Structured Query to SAP – Query Outbound Processing**

- This tutorial demonstrates how WebSphere Adapter for SAP Software uses QISS interface to retrieve data for SAP Table Kna1.

---

**Configuration prerequisites**

- Please refer to section 2.1 for configuration details.

---

**Configuring the adapter for outbound processing**

- Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

- When the WID is opened, Business Integration view is presented.

- Start the Adapter EMD by choosing: File-> New -> External Service
1. Select SAP Adapter then click Next

- Expand the “IBM WebSphere Adapter for SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and required connection. Then click next.
Select an Adapter
Select the adapter you want to use.

IBM WebSphere Adapter for SAP Software (IBM: 6.2.0.0)
IBM WebSphere Adapter for SAP Software with transaction support (IBM: 6.2.0.0)
CWIAP_SAPAdapter_Tx
  saperp05.svl.ibm.com:z10an:00
• In the required files and libraries, select files required to access SAP Server.

•

• In the processing Direction screen, select Outbound for QISS interface. Click Next to go to next screen.
Setting connection properties for the external service wizard

- To connect to SAP System the following information are necessary: hostname, username and password. Provide this information in the next wizard page. Then click Next.
Selecting the business objects and services to be used with the adapter

- Follow these steps to select the QISS business object:

  - In the Object Discovery and Selection screen, click “QISS” node below Discovered objects then Click image and enter KNA1. Then press Enter. Click to expand QISS node.”
Click on \( \text{ } \) to select Kna1 into Selected Objects.
• Accept the defaults and click OK.

• Click QISS node and then Click button again and enter ADRC. Then hit “OK”.

• Click on to select ADRC into selected Objects.

• Under select Parent Table for Table ADRC, select KNA1 from the drop down list.
• 7. Under "Map primary key columns to parent-table foreign key reference columns" choose the following from the drop down.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRNU</td>
<td>ADRN</td>
</tr>
<tr>
<td>DATE_FR</td>
<td>None</td>
</tr>
<tr>
<td>NATION</td>
<td>None</td>
</tr>
<tr>
<td>CLIENT</td>
<td>None</td>
</tr>
</tbody>
</table>

• 8. Select the defaults for "Choose columns to be selected in your query " and click OK.
9. Click Next.
Generating business object definitions and related artifacts

- Follow these steps to generate the business object definitions.
- 1. In “Configure Composite Properties” window use the default values for Object Namespace and relative folder for generated business objects

2. Click Next
3. On the Service Generation and Deployment Configuration screen enter the connection information.
4. Click Next.

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

6. If the New Integration Project screen appears, select Create a module project, then click Next.
7. In the New Module screen, type “QISSKna1AdrcSample” in the Module Name field, and then click Finish.
8. Click **Finish**. The new QISSKna1AdrcSample module is added to the Business Integration Perspective, along with all its artifacts. Save the details if prompted.

- Verify the results.
Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

- Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**. Add Project from available projects. Click Finish.
• Add the SCA module to the server.

Testing the assembled adapter application

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

• Select the service you want right click on this project, and click Test > Test Module.

• Populate values for input business objects.
• Set Operation as retrieveSapkna1

• Set verb as RetrieveAll

• Set value in Sapkna1-> CustomerNumber1=0000000001

• Add child object ADRC by right click on SapAdrc business object and add in pop menu select ‘Add Elements…’ as 1
7. Click on the ![button] button. When the Select Deployment window appears, select the WPS server to which you added the project and click on the Finish button.
8. Click OK.

9. Check the output of the service, and 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 9. Tutorial 6: Sending data from the SAP (inbound processing) using simple ALE Interface

Sending IDoc data From SAP – ALE Inbound Processing

• This tutorial demonstrates how to use External Service to generate business objects based on the IDoc, and create a module that contains WebSphere Adapter (6.2.0.0) for SAP Software. Then deploy the module in Server to the test environment of WebSphere Integration Developer (6.2). In this tutorial explains that Basic IDoc as ALEREQ01.

Configuration prerequisites

• Please refer to section 2.1 for configuration details.

Configuring the adapter for inbound processing

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File-> New -> External Service
- Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and connection. Then click next.
1. Select Inbound radio button then click Next

2. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files, Librfc32.dll/librfccm.so and
sapjcorfc.dll/lsapsapjcorfc.so files.

Setting connection properties for the external service wizard

1. To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “ALE”. Then click Next.
• Simple ALE – Figure 2: Discovering ALE from the SAP EIS

Selecting the business objects and services to be used with the adapter
- In “Object Discovery and Selection” screen.

- Click before ALE node in Discovered objects to expand this node.

- Simple ALE – Figure 3: Discover Objects and Services

- Click on “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 by Name window.
Simple ALE - Figure 4: Filter Properties for Discover IDoc from System.

- Click OK.
- Expand Discover IDOC From System node.
• Simple ALE-Figure 5: Discovered ALEs using

• Discover by Name (filtered)

• Navigate to Alereq01 and click the button.

• In the Configuration Parameters window, click OK.
• Simple ALE – Figure 6: Setting configuration parameters for the ALE selected

• ALEREQ01 has now been added to the list of business objects to be imported. Click Next.
• Simple ALE – Figure 7: Selected ALE added to the Objects to be imported

• In the “Configure Composite Properties” window, enter “bodefs” as the name of the object location, and add “Create” to the Service operations.
• Simple ALE – Figure 8: Configure Objects

• Select any of existing IDoc value to identify selected operation.
• Click Next
7. On the Service Generation and Deployment Configuration screen enter the connection information. Click next.

8. In the Service Location Properties screen, click the **New** button next to the Module field to create a new module.
9. If the New Integration Project screen appears, select **Create a module** project, then click **Next**
10. In the New Module screen, type **ALEINSample** in the Module Name field, then click **Finish**.
11. Click Finish.

12. Verify the results.
9.4 Generating Reference Bindings

- In the Business Integration Perspective of WebSphere Integration Developer, right-click the “ALEINSample” SCA module, and select Open Assembly Diagram. The Assembly Diagram window appears with the module's Export component in view.
• ALE Inbound – Figure 12: Opening the Assembly editor
2. To create a new component, click the button of java component from the “Palette”.

![ALE Inbound interface in the Assembly editor](image)

- 3. Click the palette to add the new component to the Assembly Diagram window.
- Click and drag the java component to the new component. This draws a wire from the Export component to the new component and displays the Add Wire window.
• ALE Inbound – Figure 14: ALE Inbound interface being wired to a target Component (end-point)

• 5. In the Add Wire window, click OK.

• ALE Inbound – Figure 15: Confirmation to create an export for the target Java Component (wiring)

• 6. Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
- ALE Inbound – Figure 16: Creating Java implementation for the target Component.

- 7. In the Generate Implementation window, select the package in which the Java code will be created and click **OK**. A Java file in an editor window appears.
• ALE Inbound – Figure 17: Specifying the package for the Java implementation used by the target Component.

• 8. Edit the Java file. For example, you may wish to write code to print trace and log messages.

• 9. Save the Java file

• 10. Save assembly diagram.

9.5 Deploying the module to the test environment

The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.
• Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select Add and remove projects.

![Add and Remove Projects](image)

• Add the SCA module to the server. Click “Finish” to complete this step.

---

**Testing the assembled adapter application**

• When click the finish button, then ALEINSample module project will deploy in server. You can verify from the console message.
Send Event from SAP

- Use the **WE19** transaction in the SAP client user interface to send an ALEREQ01 IDoc from the SAP instance.

- Choose radio button “Existing IDoc”. Select an existing ALEREQ01 IDoc that you want to send out.

- Select IDoc → Create from the menu.

- Set appropriate values in IDOC.

- Select “Standard Outbound Processing” button.

- Select “Continue” in the pop-up box

  a. This creates an event for the ALE inbound application.
Chapter 10. Tutorial 7: Sending data to SAP (outbound processing) using simple ALE Interface

Sending IDoc data to SAP – ALE Outbound Processing

- This tutorial demonstrates how to use External Service to generate business objects based on the IDoc, and create a module that contains WebSphere Adapter (6.2.0.0) for SAP Software. Then deploy the module in Server to create IDoc in SAP system and test the environment of WebSphere Integration Developer. In this tutorial explains that Basic IDoc as ALEREQ01.

Configuration prerequisites

- Please refer to section 2.1 for configuration details.

Configuring the adapter for outbound processing

- Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.
  
  - When the WID is opened, Business Integration view is presented.
  
  - Start the SAP EMD by choosing: File-> New -> External Service
1. Select SAP Adapter then click next
2. Next Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and click next.

3. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files.
Librfc32.dll/librfccm.so and sapjcorfc.dll/libsapjcorfc.so files.

- Click Next
- Select Outbound click next
Setting connection properties for the external service wizard

- To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “ALE”. Then click Next.
Simple ALE – Figure 2: Discovering ALE from the SAP EIS

Selecting the business objects and services to be used with the adapter

1. In the “Object Discovery and Selection” screen, under “Discover IDOC From System”. Click ⌧ next to ALE node to expand tree
• Simple ALE – Figure 3: Discover Objects and Services

• 2. Click on “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 by Name window.
• Simple ALE- Figure 4: Filter Properties for Discover IDoc from System.

• 3. Click OK.

• 4. Expand Discover IDOC From System node.
- Simple ALE-Figure 5: Discovered ALEs using Discover by Name (filtered)

- 5. Navigate to Alreq01 and click the button.

- 6. In the Configuration Parameters window, click OK.
• Simple ALE – Figure 6: Setting configuration parameters for the ALE selected

• 7. ALEREQ01 has now been added to the list of business objects to be imported. Click Next.
• Simple ALE – Figure 7: Selected ALE added to the Objects to be imported

• 8. In the Configure Composite properties window, enter “bodefs” as the name of the Folder Name
• Simple ALE – Figure 8: Configure Objects

• 9. Click Next.
10. On the Service Generation and Deployment Configuration screen enter the connection information. Click next.

11. In the Service Location Properties screen, click the **New** button next to the Module field to create a new module.
12. If the New Integration Project screen appears, select Create a module project, then click Next.
13. In the New Module screen, type **ALEOUTSample** in the Module Name field, then click **Finish**.
• Figure: Create New Module

• 14. Click Finish.

• 15. Verify the results.
Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

- Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select Add and remove 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.
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, see your SAP administrator to obtain the data.

- Client
- IdocNumber
- SenderPort
- PartnerNumberofSender
- ReceiverPort
- PartnerNumberofRecipient

How to perform this task
- In the Business Integration perspective, begin the testing procedure by right-clicking ALEREQ01 and clicking Test → Test Module.
- In the verb row under executeSapAlereq01Input, select Create from the list.
- Enter the IDoc Control record data:

  - Right-click SapAlereq01IDocBO and click Add Element
  - If you are prompted to enter the number of elements to add, select 1 and click OK.
  - Type the following values in the associated fields:

    - ReceiverPort
    - PartnerTypeOfSender
    - NameOfBasicType
    - PartnerNumberOfSender
    - NameOfTableStructure
    - Client
    - LogicalMessageType
    - PartnerNumberOfRecipient
    - SenderPort
    - IdocNumber
    - PartnerTypeOfRecipient
    -
4. Set the IDoc Data Record level property values.

- Right-click SapAlereq01DataRecord and click Add Element.
- Type the following values in the associated fields:
  - Logicalmessagetype - ALEREQ
  - Messagetype - ALEREQ
  - Right-click SapAlereq01E2aleq1 and click Add Element
  - IncludingExcludingindicator - E
  - RelationaloperatorEqNeGtLtGeLe - LT
  - Lowerlimitforfieldcontents - 0
  - Upperlimitforfieldcontents - 1

5. Click 🔄.


7. Execute the service by clicking the continue button 🔄.

8. Check the output of the service, and check the data in the EIS to ensure it matches expected values.
Chapter 11. Preparing to run through the AEP tutorial

Configuration prerequisites

- Before doing any tutorial testing, complete the following tasks:
  - Create an authentication alias.
  - Import the sample transports supplied along with the adapter into the SAP system.
  - The following is a sample list of the SAP R/3 version 4.7/ERP transport files necessary to support this 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 \J2CAComponents\SAPAdapter\AepTransports.

<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 jars from SAP Service Marketplace. They should be copied to a folder on the system where the WID EMD will be executed.
Extracting the sample files

- Replicas of the artifacts that you create when using the external service wizard are provided as sample files for your reference. Use these files to verify that the files you create with the 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.im</td>
<td>Contains the SCA im for the resource adapter.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SAPOutboundInterface.wsdl</td>
<td>Service interface to in the resource adapter.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01.xsd</td>
<td>Business Object def for the SapYxrv5b01.</td>
</tr>
<tr>
<td>AEPOutBoundSample/SapYxrv5b01BG.xsd</td>
<td>Business Object def for the business object graph.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv51000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv51000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv52000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv52000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv53000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv53000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv54000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv54000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv55000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv55000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv56000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv56000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv57000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv57000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv58000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv58000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv59000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv59000.</td>
</tr>
<tr>
<td>AEPOutBoundSample/ SapYxrv5b01Z2xrv5a000.x</td>
<td>Business Object definition for the SapYxrv5b01Z2xrv5a000.</td>
</tr>
<tr>
<td>File Path</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>SapYxrv5b01Z2xrv5b000.x</td>
<td>for the SapYxrv5b01Z2xrv5b000</td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Business Object definition</td>
</tr>
<tr>
<td>SapYxrv5b01Z2xrv5c000.x</td>
<td>for the SapYxrv5b01Z2xrv5c000</td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>DuplicateRecordFault.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>MatchesExceededLimitFault.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>MissingDataFault.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>MultipleMatchingRecordsFault.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>PrimaryKeyPairType.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>RecordNotFoundFault.xsd</td>
<td></td>
</tr>
<tr>
<td>AEPOutBoundSample/</td>
<td>Fault Schema</td>
</tr>
<tr>
<td>WBIFault.xsd</td>
<td></td>
</tr>
</tbody>
</table>

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

This tutorial demonstrates how WebSphere Adapter for SAP 6.2.0.0 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

- The following tasks must be completed for this tutorial:
  - Create an authentication alias on the server to process outbound requests using the administrative console on the server. Refer to UserGuide.
  - Create an adapter project in WebSphere Integration Developer by importing the adapter RAR file. Add external software dependencies (ex: sapjco jar files). Refer to UserGuide.
  - Configure the SAP system to work with the adapter. If you have not already imported the transports into the SAP system, import the transports. Make sure imported Function Modules are activated and Remote Enabled. Refer to the below link for more details on configuring the SAP system for running this interface.
  - Create a communication user in the SAP system, as the adapter requires this user and password info while making a connection to the SAP system.
  - Configure log/trace directories to have required capacity.
Configuring the adapter for outbound processing

- Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

- 1. When the WID is opened, Business Integration view is presented.

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

- 3. Select Adapters radio button then click Next
4. Expand the IBM WebSphere Adapter for SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and connection and click Next.

5. In the Required Files and libraries screen, select the version of SAP sapjco.jar, sapjcorfc.dll and librfc32.dll, which were downloaded previously for the specific OS.
6. Click Next

7. Select the Outbound and Click Next
Setting connection properties for the external service wizard

- To connect to SAP System the following information is necessary: username, password, Hostname System number and client. Provide this information in the next wizard page.
Click on **Advanced >>** button and set the following information: RFC Trace on, Bidi transformation, log file location and logging level, if required. Then click Next.

Selecting the business objects and services to be used with the adapter

Follow these steps to select the Yxrv5b01 (CustomerMaster) business object:
1. In the Object Discovery and Selection screen: select AEP node and expand it, select Discover IDoc From System, click on Create or edit filter button (as shown by the ellipse).
2. In the Filter Properties for ‘Discover IDoc From System’ window, select the values as shown below. Then click OK.

3. Select node Discover IDoc From System (filtered) and expand it. Select YXRV5B01 and click on > as shown by the ellipse below.
4. Opens a new window **Configuration Parameters for ‘YXRV5B01’**, Click on **Deselect All** button and then select only “KUNNR (Customer Number 1)” field. The **Customer Number 1** is the Primary Key field. Then click OK.
• 5. Click Next
6. Click on **Add** button, select **Create** in the popup window and click OK.

7. Populate the ABAP function module name, Business object namespace and Folder fields as shown.
8. Repeat the steps 16 & 17 for Update, Delete and Retrieve operation. Give the ABAP function module name as mentioned below.

- Retrieve - Y_XR_CUSTOMERMASTER_C4
- Create    - Y_XR_CUSTOMERMASTER_C1
- Update    - Y_XR_CUSTOMERMASTER_C2
- Delete    - Y_XR_CUSTOMERMASTER_C3
• These are custom Function Modules in SAP system provided as samples by the adapter installer, which were called based on the operation.

• 9. Click the NEXT>.

Generating business object definitions and related artifacts
• Follow these steps to generate the business object definitions.

• 1. On the “Service Generation and Deployment Configuration” screen, enter the connection information and J2C Authentication Data Entry.

• Select the checkbox Specify a JAAS alias security credential (default checked)

• Enter authentication alias name in J2C Authentication Data Entry field. (J2C Authentication Data Entry is the authentication alias created on the process server in the earlier sections of this document.)
2. Click on **Advanced** button, to configure SAP system load balancing or turn on the RFC trace or ABAP Debug mode. Then fill in the appropriate fields. Click Next.
3. In the Service Location Properties screen, click the **New** button next to the Module field to create a new module.
4. If the New Integration Project screen appears, select **Create a module project**, then click **Next**

5. In the New Module screen, type **AEPOBSample** in the Module Name field, and then click **Finish**.
6. Click **Finish** on Service Location Properties screen.

7. Verify the results.
8. Save the Assembly Diagram. Select **Project** menu > Clean… > Clean all projects > OK

---

Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.
1. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select Add and remove projects.

2. Select the project, Click Add and Click Finish.

Testing the assembled adapter application

Test the assembled adapter application using the WebSphere Integration Developer integration test client.
1. Select the service you want, and right click **Test > Test Module**.
2. Populate values for input business objects.

For Retrieve operation
- Operation: Retrieve (retrieveSapYxrv5b01)
- Customernumber1: 00000000802 (of BO SapYxrv5b01)
- Verb: Create or Update or Delete
• For Create operation
• Operation: Create (createSapYxrv5b01)
• Verb: Create

• Populate the following in SapYxrv5b01
• Title: Mr
• Name1: Sample Test
• City: Burlingame
• Sortfield: IB
• Customeraccountgroup: 0001
• Characterfieldoflength11: EN
• Transportationzonetoorfromwhichthegoodsaredelivered: 0000000001
• Countrykey: US

• Add an element to SapYxrv5b01Z2xrv51000, and populate the following
• Salesorganization: 0001
• Distributionchannel: 01
• Division: 01
• Shippingconditions: 01

• Add an element to SapYxrv5b01Z2xrv54000, and populate the following
• Nameofglobalcompanycode: 0001
- For Update operation
- Operation: Update (updateSapYxrv5b01)
- Verb: Update
- Populate the following in SapYxrv5b01
- CustomerNumber1: 0000000815
- Title: Mrs
### General Properties

- **Configuration:** Default Module Test
- **Module:** AEPOBSample
- **Component:** SAPOutboundInterface
- **Interface:** SAPOutboundInterface
- **Operation:** updateSapYxrV5b01

---

### Detailed Properties

#### Initial request parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateSapYxrV5bG</td>
<td>SapYxrV5b01BG</td>
<td></td>
</tr>
<tr>
<td>verb</td>
<td>verb&lt;string&gt;</td>
<td></td>
</tr>
<tr>
<td>SapYxrV5b01</td>
<td>SapYxrV5b01</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Function&lt;string&gt;</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>CustomerNumber1&lt;string&gt;</td>
<td>0000000815</td>
</tr>
<tr>
<td>Title</td>
<td>Title&lt;string&gt;</td>
<td></td>
</tr>
<tr>
<td>Centralord</td>
<td>Centralord&lt;br&gt;blockfor...</td>
<td></td>
</tr>
<tr>
<td>Expresstra</td>
<td>Expresstranslation&lt;string&gt;</td>
<td></td>
</tr>
<tr>
<td>Transstation</td>
<td>Transstation&lt;string&gt;</td>
<td></td>
</tr>
<tr>
<td>Character</td>
<td>CharacterfieldwithField</td>
<td></td>
</tr>
</tbody>
</table>
1. Execute the service by clicking the continue button.
2. Select the server. Click Finish.
3. Click OK.

- a. Retrieve

Return parameters

- b. Create
c. Update – check the EIS.

- 1. Check the output of the service, and check the data in the EIS to ensure it matches expected values.

- 2. Check the Event table for the newly added events

- Logon to SAP system, execute `/o/cwld/home_aep`, Click Management tab, Click on Current Events button, Give Date of Event, Click on Execute
Clearing the sample content

- Return the data to its original state.

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 WebSphere Adapter for SAP 6.2.0.0 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 Message Broker. After the events were 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

- Refer to the configuration prerequisites of Tutorial 8: AEP Interface Outbound processing, complete these to run this tutorial.

Configuring the adapter for inbound processing

- Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

- 1. When the WID is opened, Business Integration view is presented.

- 2. Start the EMD by choosing: File-> New -> External Service
• 3. Select SAP Adapter then click Next
4. Expand the IBM WebSphere Adapter for SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and click Next.

5. In the Required Files and libraries screen, select the version of SAP sapjco.jar, sapjcorfc.dll and librfc32.dll, which were downloaded previously for the specific OS.
6. Click Next

7. Select the Inbound and Click Next
Setting properties for the external service wizard

- To connect to SAP System the following information is necessary: username, password, Hostname System number and client. Provide this information in the next wizard page.
Click on Advanced >> button and set the following information: RFC Trace on, Bidi transformation, log file location and logging level, if required. Then click Next.
Selecting the business objects and services to be used with the adapter

- Follow these steps to select the Yxrv5b01 (CustomerMaster) business object:

  1. In the Object Discovery and Selection screen: select AEP node and expand it, select Discover IDoc From System, click on Create or edit filter button (as shown by the ellipse).
2. In the Filter Properties for ‘Discover IDoc From System’ window, select the values as shown below. Then click OK.
1. Select node Discover IDoc From System (filtered) and expand it.

Select YXRV5B01 and click on as shown by the ellipse below.
4. Opens a new window **Configuration Parameters for ‘YXRV5B01’**, Click on **Deselect All** button and then select only “KUNNR (Customer Number 1)” field. The **Customer Number 1** is the Primary Key field. Then click OK. ABAP function module name value is “Y_XR_CUSTOMERMASTER_C4”
5. Click Next

6. Click on Add button, select Create in the popup window and click OK.
7. Populate the ABAP function module name, Business object namespace, and Folder fields as shown
8. Click the **NEXT**

**Generating business object definitions and related artifacts**

- Follow these steps to generate the business object definitions.

1. On the Deployment Configuration screen, enter the connection information and J2C Authentication Data Entry.
2. Select the checkbox Specify a JAAS alias security credential (default checked)

3. Enter authentication alias name in J2C Authentication Data Entry field. (J2C Authentication Data Entry is the authentication alias created on the process server in the earlier sections of this document.)
- fig: Service Generation and Deployment Configuration

- 4. Click on **Advanced** button, to configure SAP system load balancing or turn on the RFC trace or Inbound event polling configuration. Then fill in the appropriate fields. Click Next.

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

- 6. If the New Integration Project screen appears, select **Create a module project**, then click **Next**
7. In the New Module screen, type **AEPIBSample** in the Module Name field, and then click **Finish**.
8. Click **Finish** on Service Location Properties screen.

9. Verify the results.
10. In the Assembly Diagram, Drag and drop Untyped component, make a link between SAPInboundInterface and Component1 as shown in the figure below. Right Click and select “Generate Implementation. click “OK” button
11. Right click on the Component1, choose **Generate Implementation**, then **Java**
• 12. Opens a new window, choose default package and click OK

![Generate Implementation dialog box]

• 13. Opens Component1Impl.java, you can add System out messages in each of the methods, save and close this class.

• Save the Assembly Diagram. Select Project menu > Clean… > Clean all projects > OK

---

**Deploying the module to the test environment**

• After running the external service wizard, you will have an SCA module that contains an EIS import or export. You must install this SCA module in the WebSphere Integration Developer integration test client.

• 1. Add the Server, if doesn’t exists already. Follow the instructions explained on the outbound section 3.3.

• 2. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select Add and remove projects
3. Select the project, Click Add and Click Finish.

4. After deploying the project, you should see the below message
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 and events can be posted to event table by using one of the options.

1. 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 on Resubmit button. Event will be posted to event table and gets processed by the adapter. **This option is mainly for testing purposes only.**

2. 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.
3. Click Continue
- Save the customer record. See in the WID that this event is picked up by the adapter and processed successfully.
1. 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.

2. Click Continue
- Save the customer record. See in the WID that this event is picked up by the adapter and processed successfully.
1. Check the output of the service, and check the data in the EIS to ensure it matches expected values.

---

**Clearing the sample content**

- Return the data to its original state.
  
  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:

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

- **Cause:** This is usually caused by wrong BO name or Component1Impl 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 Component1Impl 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 (inbound processing) using the SAP ALE-passthrough interface using a Generic IDoc

Sending IDoc data From SAP – ALE Inbound Processing

- This tutorial demonstrates how to use External Service to generate a Generic Object for all IDocs, and create a module that contains WebSphere Adapter (6.2) for SAP Software. Then deploy the module in Server to the test environment of WebSphere Integration Developer (6.2). In this tutorial, we use 2 IDoc types to test the Generic IDoc type – ALEREQ01 and MATMAS03.

Configuration prerequisites

- Please refer to section 2.1 for configuration details.

Configuring the adapter for inbound processing

- Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

- When the WID is opened, Business Integration view is presented.

- Start the SAP EMD by choosing: File -> New -> External Service
- Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node and connection. Then click next.

1. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files.
Librfc32.dll/librfccm.so and sapjcorfc.dll/libsapjcorfc.so files.

5. Select the “Inbound” radio button and click Next
6. Click Next

**Setting connection properties for the external service wizard**

1. To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “ALE pass-through IDoc”. Then click Next.
• ALE pass-through IDoc – Figure 2: Discovering ALE from the SAP EIS

Selecting the business objects and services to be used with the adapter

• In “Object Discovery and Selection” screen.”
• Click next to ALE node in Discovered objects to expand this node

• Simple ALE – Figure 3: Discover Objects and Services
1. Click on “Generic IDoc” and then click the “>” button to add it to the list of Selected Objects.

- ALE pass-through IDoc - Figure 4: Breaking into packets
- ALE pass-through IDoc - Figure 5: Selected the Generic IDoc

- 2. Click Next.

- 3. On the Service Generation and Deployment Configuration screen, enter the connection information. Click next
4. In the Service Location Properties screen, click the New button next to the Module field to create a new module.
5. If the New Integration Project screen appears, select **Create a module project**, then click **Next**
6. In the New Module screen, type **AlePassthroughInbound** in the Module Name field, then click **Finish**.
Figure 29

7. Click Finish.

8. Verify the results.
15.4 Generating Reference Bindings

1. In the Business Integration Perspective of WebSphere Integration Developer, right-click the “AlePassthroughInbound” SCA module, and select Open Assembly Diagram. The Assembly Diagram window appears with the module's Export component in view.
1. To create a new component, click the button of java component from the "Palette".

   • ALE passthrough-IDoc Inbound – Figure 13: ALE Inbound interface in the Assembly editor

   • 2. Click the palette to add the new component to the Assembly Diagram window.

   • 3. Click and drag the java component to the new component. This draws a wire from the Export component to the new component and displays the Add Wire window.
4. In the Add Wire window, click OK.

5. Right-click the new component and select **Generate Implementation**. This creates a Java component that will act as an endpoint.
ALE passthrough-IDoc Inbound – Figure 16: Creating Java implementation for the target Component.

6. In the Generate Implementation window, select the package in which the Java code will be created and click **OK**. A Java file in an editor window appears.
• ALE passthrough-IDoc Inbound – Figure 17: Specifying the package for the Java implementation used by the target Component.

• 7. Edit the Java file. For example, you may wish to write code to print trace and log messages.
• Only one message endpoint for all IDocs. This is the idea of the Generic IDoc type.

• 8. Save the Java file

• 9. Save assembly diagram.

9.6 Deploying the module to the test environment

• The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.
• 1. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**.

• 2. Add the SCA module to the server. Click “Finish” to complete this step

---

Testing the assembled adapter application

• When click the finish button, then AlePassThroughInbound module project will deploy in server. You can verify from the console message.

Send Event from SAP

• Use the **WE19** transaction in the SAP client user interface to send an ALEREQ01 IDoc from the SAP instance.
Choose radio button “Existing IDoc”. Select an existing ALEREQ01 IDoc that you want to send out.

- Select IDoc → Create from the menu.

- Set appropriate values in IDOC.

- Select “Standard Outbound Processing” button.

- Select “Continue” in the pop-up box

- This creates an event for the ALE inbound application.

- Run the same scenario for the MATMAS03 (or any other IDoc) as well.

- The events should reach the Java endpoint, indicated by an output on the console.

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

Configuration prerequisites

- **Note:** If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.o, librfc32.o files) to the `<WPS_INSTALL>\bin` directory (If WID is installed generally the WPS instance is installed under `<WID_INSTALL_DIR>\runtimes\bi_v6`).

- (For z/OS users) add the *.so libraries to the `<WAS_INSTALL>\lib` directory.

- 2. (Windows users) Install the msvcp71.dll and msvcr71.dll files in the Windows system path.

- 3. Copy sapjco.jar to the `<WPS_INSTALL>\lib` directory.

- (For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco.jar` to `WAS_SERVER_ONLY_server_region_classpath`
• Also you need this jar while running EMD wizard.

• Where <WPS_INSTALL> represents the WebSphere Process Server installation directory.

---

Configuring the adapter for outbound processing

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing: File-> New -> External Service
3. Select SAP Adapter then click Next

4. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter node. Further select the Connection that matches the
SAP Server and connection credentials. Then click next.

6. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files.
Librfc32.dll/librfccm.so and sapjcorfc.dll/libsapjcorfc.so files.

6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Outbound” radio button, then click “Next >” button.

- To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI”. Then click next.
Select BAPI as the interface

Selecting the business objects and services to be used with the adapter

2. Under “Object Discovery and Selection”, click on RFC node. Then click on the “” button.
• Figure: Object Discovery and Selection

• 2. Enter Z_ASYNCBAPI_1 (the name of the BAPI in SAP plus an asterisk as a wild card character) in the Filter Properties for Discover by Name window.
- Figure: Filter Properties for RFC
- Click “OK”.
- Expand “RFC” node.
- Figure: Retrieved BAPIs’ based on search criteria

- Select the “ZASYNCBAPI_1” from the metadata tree.

- Click on the “>” button. Clicking this should bring up the Configuration Parameters for the “ZASYNCBAPI_1”. This screen has 2 fields. The first field “Use SAP field names to generate attributes names” may be checked if attribute names need to be generated using SAP field Names. Second element
helps to select any optional parameter that the user might be interested in adding to the business objects. Click “OK”

- Figure: Setting configuration parameters for the BAPI selected

- 7. Click “Next”.

Generating business object definitions and related artifacts
• Follow these steps to generate the business object definitions.

• In the “Configure Composite Properties”, enter the business Object name for the service description. Next step is to associate a operation with the Business Object. To do that, select a service operation and then select Business Object in the next drop-down. Also the user can enter the name of the folder where the business objects are created. Click “OK” button to go back to “Object Discovery and Selection” screen.

• Choose “Asynchronous Queued RFC” for “SAP Remote Function Call(RFC) type”

• Choose the configured Queue name on the SAP server. “HENRYSERVER” is chosen on this machine.
- Figure: Configure Composite Properties

- 2. Click “Next”.
3. On the Service Generation and Deployment Configuration screen enter the connection information. Click next. Please Note that J2C authentication name should
be "<server node name>/ SAP_Auth_Alias"

- Figure: Service Generation and Deployment Configuration
• Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply enter the username and password used to login in to the SAP.

• 4. Enter the username and password and click next.

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

• 6. If the New Integration Project screen appears, select Create a module project, then click Next.

• 7. In the New Module screen, type qRFC_BAPI_outbound in the Module Name field, and then click Finish.
8. Click Finish.

9. Verify the results.
Deploying the module to the test environment

- The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

- 1. Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**. You should see the project listed under “Available projects”.

- Figure: Artifacts created after running the qRFC BAPI Module
2. 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.

- Select the qRFC_BAPI_OUTBOUND, right click and select Test > Test Module from the pop-up menu.

- Enter values as shown in the following figure.

  - CustomerNumber1 0000000001

- 3. Click on the Continue button. When the Select Deployment window appears, select the WPS server to which you added the project and click on the Finish button.

- 4. Click on OK
• Check the output of the service, and check the data in the EIS to ensure it matches expected values.

Clearing the sample content

• Return the data to its original state.

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

- Following sections explain inbound scenarios for the BAPI interface.

Configuration prerequisites

- **Note**: If you have already done this step for earlier scenario on your machine, you don’t need to repeat it you can skip this step and move to 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 dependencies libraries (sapjcorfc.dll, librfc32.dll, or sapjcorfc.so, librfc32.so or sapjcorfc.o, librfc32.o files) to the `<WPS_INSTALL>\bin` directory (If WID is installed generally the WPS instance is installed under `<WID_INSTALL_DIR>\runtimes\bi_v6`).
  - (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.
  - Copy sapjco.jar to the `<WPS_INSTALL>\lib` directory.
• (For z/OS users) add `${WAS_INSTALL_ROOT}/lib/sapjco.jar` to `WAS_SERVER_ONLY_server_region_classpath`.

• Also you need this jar while running EMD wizard.

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

Configuring the adapter for inbound processing

• Run the external service wizard to specify business objects, services, and configuration to be used in this tutorial.

• When the WID is opened, Business Integration view is presented.

• Start the SAP EMD by choosing File -> New -> External Service.
5. Select SAP Adapter then click Next

6. Expand the “IBM WebSphere Adapter SAP Software (IBM: 6.2.0.0). Select the CWYAP_SAPAdapter_Tx node. Further select the Connection that matches
the SAP Server and connection credentials. Then click next.

7. In the Connector Project Settings Screen, provide the path for “sapjco.jar”. Also provide the path for the two System Library files. Librfc32.dll/librfccm.so and
sapjcorfc.dll/libsapjcorfc.so files.

6. Click Next

Setting connection properties for the external service wizard
In “Processing Direction” screen, select “Inbound” radio button, then click “Next >” button.

- To connect to SAP System the following information is necessary: username, password, Hostname. Also enter the system Number, Client of SAP System you would like to connect to. Language code is set by default to English and can be changed using the “select” button. Codepage is by default set to “1100” and can be changed to other values available in drop-down. SAP interface name should be selected as “BAPI”. Then click next.
Selecting the business objects and services to be used with the adapter

- 1. Under “Object Discovery and Selection”, click on RFC node. Then click on the “…” button.
- **Figure**: Object Discovery and Selection

- 2. Enter Z_ASYNCBAPI_1 (the name of the BAPI in SAP plus an asterisk as a wild card character) in the Filter Properties for Discover by Name window.
• Figure: Filter Properties for RFC

• 3. Click “OK”.

• 4. Expand “RFC” node.
Figure: Retrieved BAPIs’ based on search criteria

- 1. Select the “Z_ASYNCBAPI_1” from metadata tree.

- Click on the button. Clicking this should bring up the Configuration Parameters for the “Z_ASYNCBAPI_1”. This screen has 2 fields. The first
field “Use SAP field names to generate attribute Names” may be checked if attribute names need to be generated using SAP field Names. Second element helps to select any optional parameter that the user might be interested in adding to the business objects. Click “OK” button to go back to “Object Discovery and Selection” screen.

Figure: Setting configuration parameters for the BAPI selected

2. Click “Next”. 
Generating business object definitions and related artifacts

- Follow these steps to generate the business object definitions.

- 1. In the “configure Composite properties” Screen, associate the RFC-enabled function name with an end-point operation. Enter the Business Object namespace. Also the user can enter the name of the folder where the business objects are created.
• Figure 8: configure Composite properties

• 2. Click “Next”.
3. On the Service Generation and Deployment Configuration screen enter the connection information. Click next.
Service Generation and Deployment Configuration

Service operations
If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties
 Specify a Java Authentication and Authorization Services (JAAS) alias security credential.
 J2C Authentication Data Entry: widNode/SAP_AUTH_ALIAS
 Deploy connector project: With module for use by single application

Specify the settings used to connect to SAP software at runtime:
 Connection properties: Specify connection properties

Connection properties
 SAP system connection information
 Host name: saperp04.svLibm.com
 RFC program ID: HENRYSERVER
 Gateway host: saperp04.svLibm.com
 Gateway service: sapgw00
 Client: 200
 Language code: EN (English)
 Code page: 1100
 System number: 00

The user name and password will not be encrypted and will be stored as plain text.

User name: songhan
Password: *********

Advanced >>
• Figure: publishing Object Configuration Properties

• Here either the user can enter Authentication Alias already created using the “administrative Console” of the WPS or simply 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.

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

• If the New Integration Project screen appears, select Create a module project, then click Next
5. In the New Module screen, type **qRFC_BAPI_inbound** in the Module Name field, and then click Finish.
6. Click Finish.

7. Verify the results.

Figure: Artifacts generated after SCI module is run.

### 7.2.4 Generating Reference Bindings

- In the Business Integration Perspective of WebSphere Integration Developer, right-click the “qRFC_BAPI_inbound” SCA module, and select Open Assembly Diagram. The Assembly Diagram window appears with the module’s Export component in view.
To create a new component, click the button of java component from the “Palette”.

- BAPI Inbound – Figure 13: BAPI Inbound interface in the Assembly editor

- Click the palette to add the new component to the Assembly Diagram window.

- Click and drag the java component to the new component. This draws a wire from the Export component to the new component and displays the Add Wire window.

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

- 5. In the Add Wire window, click OK.
ALE Inbound – Figure 15: Confirmation to create an export for the target Java Component (wiring)

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

```java
/**
 * Method generated to support implementation of operation "emitCreateAfterImageSap2Asyncbapi1Wrapper"
 * named "SAPInboundInterface".
 * 
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */

public DataObject emitCreateAfterImageSap2Asyncbapi1Wrapper(
    DataObject emitCreateAfterImageSap2Asyncbapi1WrapperInput) {
    try {
        System.out.println(AdapterBOUtil.serializeDataObject(emitCreateAfterImageSap2Asyncbapi1WrapperInput));
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return emitCreateAfterImageSap2Asyncbapi1WrapperInput;
}
```

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

- Edit the Java file. For example, you may wish to write code to print trace and log messages or Data Object

- Save the Java file
• Save assembly diagram.

#### Deploying the module to the test environment

• The result of running the external service wizard is an SCA module that contains an EIS import or export. Install this SCA module in WebSphere Integration Developer integration test client.

• Add the module you created earlier to the server using the server panel in WebSphere Integration Developer. Right-click on the server, and then select **Add and remove projects**. The project will be seen under “Available projects”.

![Add and Remove Projects](image)
• Figure: Available projects

• Click on the Add button and it should be transferred into the configured projects.

• Figure: Configured projects

• Add the SCA module to the server. Click finish to complete it

Testing the assembled adapter application
To enable qRFC communications, the destination must be defined in the sending system. To define a sending system, use transaction SM59.

Then, you need to register a RFC destination for qRFC destination. 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.

- Next, you need to register a queue name.
  Invoke transaction SMQR
  In the first screen, you will see a list of registered queue names.

```plaintext
<table>
<thead>
<tr>
<th>C1</th>
<th>Queue name</th>
<th>Type</th>
<th>Mode</th>
<th>Max. Runtime</th>
<th>Attempts</th>
<th>Pause</th>
<th>Destination with LOGON Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>ABC</td>
<td>R</td>
<td>D</td>
<td>60</td>
<td>0</td>
<td>360</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>ALBERTQUEUE</td>
<td>R</td>
<td>D</td>
<td>60</td>
<td>0</td>
<td>360</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>SARATHQUEUE</td>
<td>R</td>
<td>D</td>
<td>60</td>
<td>0</td>
<td>360</td>
<td></td>
</tr>
</tbody>
</table>
```

Click on 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_INBOUND1:

**"Local Interface:
**
IMPORTING
**" VALUE(CUSTOMERNO) TYPE BAPICUSTOMER_ID-CUSTOMER
**" VALUE(RFCDEST) TYPE RFCDEST-RFCDEST
**" VALUE(QUEUE_NAME) TYPE TRFCQOUT-QUEUE
**

DATA: BAPI_CUSTOMERADDRESS LIKE BAPICUSTOMER_ADDRESS,
    BAPI_CUSTOMERGENERALDETAIL LIKE BAPICUSTOMER_KNA1,
    BAPI_CUSTOMERCOMPANYDETAIL LIKE BAPICUSTOMER_O5,
    BAPI_RETURN LIKE BAPI_RETURN,
    BAPI_CUSTOMERBANKDETAIL LIKE BAPICUSTOMER_O2 OCCURS 0 WITH HEADER LINE.

CALL FUNCTION 'TRFC_QUEUE_INITIALIZE'.

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

* Making QRFC BAPI call
CALL FUNCTION 'TRFC_SET_QUEUE_NAME'
  EXPORTING
    QNAME = QUEUE_NAME.

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

<table>
<thead>
<tr>
<th>Test Function Module: Initial Screen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test for function group: ZSANTEST</td>
</tr>
<tr>
<td>Function module: Z_QRFC_ABI_INBOUN1</td>
</tr>
<tr>
<td>Upper/Lowercase:</td>
</tr>
<tr>
<td>RFC target sys: SARAMASERVER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Import parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMERNO</td>
<td></td>
</tr>
<tr>
<td>RFCDEST</td>
<td></td>
</tr>
<tr>
<td>QUEUENAME</td>
<td></td>
</tr>
</tbody>
</table>

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

• Invoke the Outbound queue monitor using transaction SMQ1

<table>
<thead>
<tr>
<th>Program</th>
<th>Edit</th>
<th>Goto</th>
<th>System</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

qRFC Monitor (Outbound Queue)

<table>
<thead>
<tr>
<th>Client</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue Name</td>
<td></td>
</tr>
<tr>
<td>Queue Destination</td>
<td></td>
</tr>
<tr>
<td>Waiting Queues Only</td>
<td></td>
</tr>
</tbody>
</table>

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

• Double click on the queue name to see the LUWs.

<table>
<thead>
<tr>
<th>LUW</th>
<th>FUNCTION MODULE</th>
<th>MODULE NAME</th>
<th>QUEUE</th>
<th>CLIENT</th>
<th>DATE</th>
<th>TIME</th>
<th>STATE</th>
<th>PRIORITY</th>
<th>MEM</th>
<th>DURATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Z_ABCDO1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Z_ABCDO1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Z_ABCDO1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Double click on Execute LUW(F6) to execute the logical unit of work.
The SAP system will send the LUW to the destination once its alive.

Clearing the sample content

- Return the data to its original state.

  Nothing is required to clean up after this tutorial.
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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows 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).