Oracle Service Bus - MySAP adapter synchronous events with SAP R/3

An Oracle White Paper
April 2009
INTRODUCTION

The Oracle Service Bus (OSB) Adapter for MySAP is part of the OSB Adapter install. The OSB Adapter for MySAP can be deployed as both a web service servlet as well as JCA 1.5 Resource Adapter. The Adapter Framework is used for the bidirectional integration of the JCA 1.5 resource adapters with Oracle Service Bus. Adapter framework is based on open standards and acts as a pseudo JCA 1.5 container for deployment on top of J2EE JCA 1.0 containers. It manages the life-cycle (start/stop/endpoint activation/endpoint deactivation) as well as the message-inflow system contracts for receiving adapter events and forwarding it to the OSB’s business and proxy services.

The OSB Adapter for MySAP supports two ways of receiving SAP events. It can be either by Remote Function Call (RFC) or Intermediate Documents (IDOC). During event processing, the adapter receives RFCs and IDocs directly from SAP. SAP Message Objects define SAP events and they are published from SAP. The SAP Integration Broker is used for configuring a process for publishing SAP Message Objects to other 3rd party applications. The OSB Adapter for MySAP can subscribe to SAP Message Objects.

OSB Adapter for MySAP supports events either asynchronously or synchronously. In the synchronous event model, the enterprise application would publish an output for an event and would wait till it receives a response for the publication of the message. For example, when a user adds a customer in the banking application, it initiates an event by sending a request from the banking application to check the credit score of that customer with creditcheck application. The creditcheck application would respond back with the credit score for the customer. The banking application will receive the credit score details and will update the customer records in the banking application. All these constitute one single transaction.
In the case of synchronous events with SAP R/3, when SAP publishes an event, it becomes the client. The OSB MySAP adapter acts as a server by receiving the request and does the processing by invoking other applications and sends the response back to SAP. Till then, the SAP R/3 application waits to receive the response back from the adapter. The RFC objects of SAP support synchronous events and the IDOCs do not support the synchronous events.

This document describes the details on how the MySAP adapter works synchronously. In this article, two SAP R/3 systems have been configured to show how they communicate with each other using synchronous events. One SAP R/3 system would act as a source system which initiates an event and other SAP R/3 system would act as a target system from where details are extracted. This scenario can be tried with any other EIS also. But, the source system should be always a SAP system. (i.e) the synchronous event origination can be done only from the SAP system. Target system can be a non-SAP system which can be other ERP's such as Siebel, PeopleSoft or JD Edwards or etc.,

This document provides step by step details on how to receive the event notification from SAP backend to BPEL through MySAP adapter and replies back after extracting the details from another system. The document has been developed using the 10gR3 OSB & Adapters installed on the windows platform and SAP R/3 version 4.7D. Hence, if you are using a different platform, then correlate the terms and commands specific to that operating system by referring the user guides of the specific component.

Note: Depending on the release or service pack installed, certain RFCs may not exist in your particular SAP R/3 system. Therefore, the examples included in this documentation may not be relevant to your system. If this is the case, you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your SAP application environment.

**Configuring the OSB adapter for MySAP**

The OSB Application Explorer is used to configure the OSB Adapter for MySAP. WSDL files are created for both JCA Outbound (Request-Response service – BPEL invoke) and JCA Inbound (Event Notification – BPEL receive) interactions and saved to a local directory. These WSDLs are exposed to the JDeveloper via the WSIL servlet. The JDeveloper based design-time wizard has a WSIF browser that is launched from the BPEL Partner Link activity to explore the WSDLs generated by the OSB Adapter Application Explorer.
Creation of the JCA configuration

Start the Application Explorer. On top left, right click on configuration and select New. Following window shows up. Enter the details as shown below and select the OK button:

![New Configuration window]

New configuration window shows up. Select JCA for Service Provider from the drop down list. Enter the adapter installed folder for the Home.

A new configuration by name jca_sample has been created and that will be shown in the Application Explorer as shown below:
Creation of the Adapter Target

Select *jca_sample*, right click and select connect to connect to the jca configuration. That will show two new nodes *Adapters* and *Events*.

Create a new target for the MySAP Adapter. This is the source system from which events will be triggered. For identification purposes, we can call this as sap1.
Enter the values under the **User** tab.

![User tab screenshot](image1)

Enter the values for the **system** tab.

![System tab screenshot](image2)

Under the **Advanced** and **Security** tab, leave the default values.
After creating the target, click on the target. Enter the password on the right pane, then right click on the target and select **connect** to connect to the target.

After connecting to the target, nodes for **Business Object Repository**, **Remote Function Modules** and **IDOCs** will be shown in the Application Explorer.

Create one another target for the MySAP adapter for a different SAP R/3 system. Follow the steps in the Creation of the Adapter Target section. This is the target SAP system, which will respond for the requests from the source SAP system. For identification purpose, we can call this target system as sap2. In real life scenario, the target SAP system could be different EIS other than SAP.
Configuration of the Channel

Channel establishes the connection between SAP R/3 and adapter and facilitates to receive the SAP R/3 published messages.

Select the **Events** tab and then select the **MySAP** node within the **events** node. Note that the channel configuration is for the source SAP System (sap1 target in the Application explorer).

Select the **Channels** under the **MySAP** node and then right click on the **channel** to **Add Channel**.
Add Channel window shows up.

Provide the following information:

a. Enter a name for the channel, for example, MySAP_Channel1.
b. Enter a brief description (optional).
c. From the Protocol list, select Application Server - mySAP.

Then click the Next button at the bottom of the window.

Following window shows up. Enter the details under the User tab as shown below:
Enter the details under the **system** tab as shown below:

![Edit Channel](image1.png)

Leave the default values under the **security** tab. Under the **Advanced** tab, enter the values as shown below:

![Edit Channel](image2.png)

Then click the **OK** button.

The channel that was created shows as like below in the Application Explorer. The channel appears under the channels node in the left pane. An X over the icon indicates that the channel is currently stopped.

![Application Explorer](image3.png)

Do not start the channel, as it is managed by BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before runtime.
Generating WSDL for RFC inbound

After you create a channel and verify that it is not started, you must generate WSDL for the event using Application Explorer.

Go to the adapter target (sap1) and select the Remote Function Modules (RFC) node.

RFCs node explode and the list of objects shows up:
Scroll down and select the **Financial Accounting -> 0002 – Company Code Business Object – BAPI_COMPANYCODE_GETLIST**.

Right click and select the **Export Inbound JCA Service (Event)** option.
Window for *Export WSDL* shows up. Name of the WSDL shows up automatically. Select following:

- **Location**: specify the OSB location for publishing the Inbound wsdl
- **Host**: host name where the wsdl to be published
- **Port**: port where WLS is listening
- **User**: weblogic
- **Password**: weblogic
- **Element form Default**: Qualified
- **Channel**: use the one that was created in step 10.
- **Port**: No Port
- **Validation**: Do not select anything

Then click the *OK* button. Now, the WSDL would be created and will be stored in a folder accessible for the JDeveloper.
Connect to the second SAP target (SAP2). This is the target SAP system.

After connecting to the target, you will see the following:

Expand the Remote Function Modules to select the object *Financial Accounting* → 0002 – Company Code Business Object – *BAPI_COMPANYCODE_GETLIST*.
Right click and select *Export Outbound JCA Service*(Request / Response) to generate the WSDL for the outbound operation.

<table>
<thead>
<tr>
<th>Test Run</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Schema(s)</td>
<td></td>
</tr>
<tr>
<td>Create Event Port</td>
<td></td>
</tr>
<tr>
<td>Create Inbound JCA Service(Event)</td>
<td></td>
</tr>
<tr>
<td>Create Outbound JCA Service(Request,Response)</td>
<td></td>
</tr>
<tr>
<td>Export Inbound JCA Service(Event)</td>
<td></td>
</tr>
<tr>
<td><strong>Export Outbound JCA Service(Request,Response)</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Export WSDL** dialog box opens and select the qualified WSDL option

- **Location** - specify the OSB location for publishing the Inbound wsdl
- **Host** - host name where the wsdl to be published
- **Port** - port where WLS is listening
- **User** - weblogic
- **Password** - weblogic

Click the **OK** button. The created WSDL can now be accessed from the JDeveloper.

Make sure that the channel is stopped. It should not be up and running. Then Close the Application Explorer.

Recycle the OSB. This is required for the OSB to read the adapter repository to understand the new changes.
Configuring Synchronous Service in Oracle Service Bus

Creation of Wsdl Type Business Service
Login to the OSB console. Using the Project explorer create folders by name “Business service” and “Proxy service”.
Select the Business Service folder and Select the Business Service from the Create Resource drop down list under Resources.

Enter a name for the Service name. Enter the value for Description as required.
Select the Service type as WSDL web service and click on Browse button.
From the list of WSDLs, Select the outbound wsdl `BAPI_COMPANYCODE_GETLIST_INVOKE` and click the Submit button.

Select `jcabinding` for bindings under Select WSDL definitions and click the Submit button.
Now, you will be taken back to the first screen of the Create a Business Service. Click the Next button now.

Select jca for the Protocol from the drop down box. Leave Load Balancing Algorithm to the default. For the Endpoint URI, enter the jndi value eis/OracleJCAAdapter/DefaultConnection and click on Add button. The value for the jndi has been taken from the outbound wsdl. Click the Next button now.
Make sure that the field *Always use configuration from JCA WSDL* is selected. The *connection mode* is set to *Managed*. Then click the *Next* button.

Verify all the details in the page. Then, click the *Save* button.
Creation of Wsdl Type Proxy Service – Source SAP system

From the **Project Explorer**, select the folder **Proxy service**. Select the **Proxy Service** from the **Create Resource** drop down list.

Enter a name for the **Service Name** and enter appropriate details for the **Description** field. Select the **WSDL Web Service** for **Service Type** and click the **Browse** button.
From the list of WSDLs, select the Inbound Wsdl BAPI_COMPANYCODE_GETLIST_RECEIVE and click the Submit button.

Select jcbinding for Bindings under the Select WSDL definitions and click the Submit button.

Now, you will be taken back to the first screen for the Proxy service configuration. Click the Next button.
Select `jca` for Protocol. For the **Endpoint URI**, enter the jndi value `eis/OracleJCAAdapter/DefaultConnection` and click on **Add** button. The value for the jndi has been taken from the inbound wsdl. Click the **Next** button now.

Make sure that the field **Always use configuration from JCA WSDL** is selected. The **connection mode** is set to **Managed**. Then click the **Next** button.
Leave the default value and click on Next button.

Click the Next button.

Verify the details and click the Save button.
Configuring the Pipeline

Select the Proxy Service that has been created already and click on the Action for *Edit Message flow*.

A new page opens for *Edit Message flow*. Select the proxy service and right click to *Add Route* in the Proxy Service.

Select the *RouteNode* and right click to select *Edit Route*. 
Select the **Add an Action** and right click to select **Communication -> Routing**.

Now, click the `<Service>` link to select the wsdl based business service created for outbound wsdl (Target SAP system).

Select the Business Service that was created earlier (section **Creation of Wsdl Type Business Service**) and click the **Submit** button.
Now, select the operation as BAPI_COMPANYCODE_GETLIST from the drop down list for **invoking**.

Select **Add an Action** in the **Request Action** and right click to select **Reporting -> Log**.

Select the link for **Expression**.
In the XQuery Expression Editor Page type $body and click on Validate button to validate the action and Click on Save button to save the action.

Select **Error** for the *at severity level* from the drop down list.

Select **Add an Action** in the **Response Action** and right click to select **Reporting** -> **Log**.
The screen will look as like shown below:

Now, click the **Validate** and then the **Save** button.
Now, select the **save** button to save the *Edit Message Flow*.

Now click the **Activate** button under the *Change Center* and click **Submit** button in the OSB console to save the changes.
SAP Event Configuration

An event in SAP is defined as an occurrence of a status change in an object. The event is created when the relevant status change occurs. Either the user or SAP must implement event creation. An event is created from a specific application program (the event creator) and then "published" system-wide.

An unlimited number of receivers can respond to the event with their own "response mechanisms". An event is usually defined as a component of an object type.

SAP pseudo events are events that are not published by the SAP Event manager, but are called from an ABAP program or Remote Function call (using the Destination parameter).

Client and Server Programs

Remote Function Call (RFC) programs for non-SAP systems can function as either the caller or the called program in an RFC communication. The two types of RFC programs are:
- RFC Client
- RFC Server

The RFC client is the instance that calls the RFC library to execute the function that is provided by an RFC server. The functions that can be executed remotely are called RFC functions, and the functions provided by the RFC API are called RFC calls.

SAP Gateway

The SAP Gateway is a secure application server. No connections are accepted unless they were pre-registered previously from the SAP presentation Client. A server connection presents itself to the Gateway and exposes a Program Identifier. If the Program Identifier is found in the list of registered Program IDs, the Gateway server then offers a connection to the server, which accepts a connection.

The Program ID then is linked with an RFC Destination within SAP, which enables SAP Function Modules and ALE documents (IDocs or BAPI IDocs) to be routed to the destination. The RFC Destination functions as a tag to mask the Program ID to SAP users. An RFC server program can be registered with the SAP Gateway and wait for incoming RFC call requests. An RFC server program registers itself under a Program ID at an SAP Gateway and not for a specific SAP system. In SAPGUI, the destination must be defined with transaction SM59, using connection type T and Register Mode. Moreover, this entry must contain information on the SAP Gateway where the RFC server program is registered.

Registering Your Program ID in SAPGUI
To enable your SAP system to issue the following calls or interfaces to the SAP event adapter, you must register your program ID under an RFC destination.

- Remote Function Calls (RFC)
- Business Application Programming Interfaces (BAPI)
- Intermediate Documents (IDOC)

The RFC destination is a symbolic name (for example, BPELDEST) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

### How to Register Your Program ID

To register your program ID:

1. Launch the **SAP Workbench** and logon to the SAP system.
2. Execute the **SM59** transaction or Select **Tools, Administration, Administration, Network**, and then **RFC destination**.

The Display and maintain RFC destinations window opens and displays a list of connections and drivers you can manage as shown in the following image.

Select **TCP/IP connections**.
Click **Create**.

The RFC Destination window opens and displays fields where you provide information about the RFC destination as shown in the following image.

![RFC Destination window](image)

- In the **RFC destination field**, type a name, for example, BPELDEST. The value you type in this field is case-sensitive.
- In the **Connection type field**, type **T** (for destination type, TCP/IP).
- In the **Description** field, type a brief description.

Click **Save** from the tool bar or select **Save** from the Destination menu.

The RFC Destination BPELDEST window opens as shown in the following image.

![RFC Destination BPELDEST window](image)

- For the Activation Type, Select the radio button for **Registered Server Program**.
- In the **Program field**, type BPELID.

Click **Save** from the tool bar or select **Save** from the Destination menu.

Ensure your channel is running.

To verify that the SAP system and the OSB Adapter for mySAP ERP are communicating, click **Test connection**.
RFC Function module should have been enabled for Remotely-callable

Access the menu SE37 from SAP GUI. In the SAP Server, the SE37 transaction enables you to send RFCs (Remote Function Calls) or BAPIs (Business Application Programming Interfaces) to any RFC destination.

Enter BAPI_CUSTOMER_GETLIST for the Function module and select the Display button.

Following screen is shown. Select the tab Attributes.
Under the **Processing type**, you would see the following:

<table>
<thead>
<tr>
<th>Processing type</th>
<th>General Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal function module</td>
<td>Person Responsible</td>
</tr>
<tr>
<td>Remote-enabled module</td>
<td>Last changed by</td>
</tr>
<tr>
<td>Update module</td>
<td>Changed on</td>
</tr>
<tr>
<td>Start immediate, no restart</td>
<td>Package</td>
</tr>
<tr>
<td>Start delayed</td>
<td>Program name</td>
</tr>
<tr>
<td>Cold run</td>
<td>INCLUDE name</td>
</tr>
<tr>
<td></td>
<td>Original language</td>
</tr>
<tr>
<td></td>
<td>Released on</td>
</tr>
</tbody>
</table>

Make sure that the radio button for the **Remote-enabled module** is selected.
Triggering the event

Login to SAP GUI.

You will see the following screen of SAP.
Select the function **SE37** from the menu.

You would see the following screen. Select BAPI_COMPANYCODE_GETLIST for the **Function module**.

Select the **single test (F8)** in the below options.

That will take you to the following Function module test screen. Enter the appropriate **RFC target system** value as shown below:
Select the *Execute (F8)* button in the test function module screen.

### Test Function Module: Initial Screen

<table>
<thead>
<tr>
<th>Export parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETURN</td>
<td>🍔</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMPANYCODE_LIST</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>🍔 0 Entries 🍔 128 Entries</td>
</tr>
</tbody>
</table>

You will see the *Result* for the execution.

Click the value for the *Result* in the screen.

<table>
<thead>
<tr>
<th>COMPANYCODE_LIST</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>🍔 0 Entries 🍔 128 Entries</td>
</tr>
</tbody>
</table>

This will show the results of the operation.

<table>
<thead>
<tr>
<th>COMP</th>
<th>COMP_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>R380</td>
<td>IDES Retail INC. US</td>
</tr>
<tr>
<td>R130</td>
<td>IDES Retail GmbH</td>
</tr>
<tr>
<td>IN31</td>
<td>India Model Company, IN</td>
</tr>
</tbody>
</table>

This completes the article for working with the synchronous events with MySAP adapter with the Oracle Service Bus.
CONCLUSION

Working with SAP R/3 synchronous events through OSB MySAP adapter is simple and elegant. Adapter provides a tightly coupled way of integrating SAP R/3 events with OSB.

REFERENCES

Oracle® Service Bus for MySAP User Guide 10g Release 3