The Oracle AS Adapter for AQ interfaces with the Business Event System to receive Oracle Applications events in real-time. The Oracle AS Adapter for Database can invoke any stored procedure or function. The Oracle AS Adapter for Siebel interfaces with Siebel Business Objects and Services to post and query Siebel data. This tutorial shows the integration of Oracle Applications with Siebel using the AQ, Database and Siebel adapters. The business process is triggered by the arrival of Oracle Applications business event and it inturn invokes Peoplesoft CustAccount Insert Business Object interface.
Overview

The Oracle as Adapter for Peoplesoft can be deployed as both a Web Service Servlet (referred to as Business Services Engine – BSE for short) as well as JCA 1.5 Resource Adapter. The Adapter operations can be accessed either via JCA-CCI API or by WSDL/SOAP or via WSDL/JCA extension. This example showcases the Web Service deployment of the Peoplesoft Adapter and the Peoplesoft Adapter operations are expressed as WSDL/SOAP.

The Oracle AS Adapter for AQ and Database are deployed as JCA 1.5 Resource Adapter and the Adapter Framework is used for the bidirectional integration of the Adapter with BPEL Process Manager. Adapter FW is based on open standards and employs the Web Service Invocation Framework (WSIF) technology for exposing the underlying JCA Interactions as Web Services.
There are 3 steps involved in receiving an event from Business Event System (BES) as the event published by the BES contains key fields only and does not include the entire payload. A subsequent API/VIEW call is required to fetch the relevant data. The 3 steps in this demo scenario are (1) AQ Adapter that subscribes to WF_DEFERRED queue (2) transformation step to extract the key fields (3) DB Adapter stored procedure call to get the relevant data.

**Configuration and testing of Siebel Service**

Install the Oracle AS Adapters CD on top of Oracle Application Server home. Copy the relevant Peoplesoft third party library – psjoa.jar to the `<APP_SERVER_HOME>/adapters/application/lib` folder. The Oracle AS Application Explorer is used to configure the Peoplesoft Adapter.

Create a Web Service repository project “IBSE” as shown in the following figure

![Application Explorer](image)

Select the “IBSE” project and select “connect” [Right-click to show options]. Connect to the Adapter. The following window shows up.

![Application Explorer](image)

Expand the Adapters and Event icons.
Connect to Siebel instance. Browse Siebel Business Objects and Select “Contact/Account” object.

Create a Web Service for insert operation as shown in the above figure. Test the Web Service from the Adapter BSE pages. Go to http://<host:port>/ibse/IBSEServlet. The Web Service in
this example is created under “test” domain. Navigate to the Siebel_Contact_Account Web Service under test domain. Test the WebService as shown in the figure.

Oracle Integration Business Service Engine
Listening on IBEServlet

The following licenses are available on IBEServlet:

- **1VP**
  The 1VP License is installed by default. It is used to install predefined Integration Business Services Engine Services.
- **test**
  The test License is installed by default. It is used to test Integration Business Services Engine Services.
- **production**
  The production License is installed by default. It is used for production purpose.

Oracle Siebel_Contact_Account
An Integration Business Service

The following operations are supported. For a formal definition, please review the Service Description.

- **insert**
- **update**

The Payload is shown in the below figure.

```xml
<Siebel location="/s80/Contact/Account/index">  
  <insert>
    <Account_spcStatus>Inactive</Account_spcStatus>
    <City>Redwood Shores</City>
    <State>CA</State>
    <Street_spcAddress>300, Oracle Way</Street_spcAddress>
    <Employees>3456</Employees>
    <Location>Oracle_iWay HQ</Location>
    <Name>Test 00</Name>
    <Currency_spccode>USD</Currency_spccode>
  </insert>
</Siebel>```

Select “Invoke” button for testing the Web Service.
This completes the configuration and testing of Siebel Web Service. The next set of steps explain the configuration of the AQ Adapter Service to receive the BES event from Oracle Applications.

**Configuration of BES service**

The Business Event System is used for bi-directional integration with Oracle Applications. It exposes key business integration points as Business Events. Subscriptions to the business events provide the means for non-invasive asynchronous bi-directional integration. It uses the Oracle Advanced Queuing (AQ) infrastructure to communicate business events between various modules of the Oracle Applications and to 3rd party applications as well. The Business Integration Points write to specific Queues (“WF_DEFERRED” queue) upon generation of an
event and the consumers of this event have to implement the subscription code to receive the event. Only the key fields of the event are written to the Queues. A subsequent request-response is needed to obtain the entire event. The below scenario shows an AQ Adapter listening to a particular event on “WF_DEFERRED” queue and a DB Adapter Stored Procedure invoke call to retrieve the entire data associated with the above event. The event shown is “APPS:oracle.apps.ar.hz.Organization.create” and is associated with the creation of a new “Organization” object. The Stored Procedure invoked is “XXBPEL_PARTY_PUB.GET_PARTY” and is a custom Procedure call to fetch the “Party” object which is a base class of the “Organization” object.

Open the JDeveloper IDE that comes along with the BPEL Process Manager product. Create an empty BPEL project and name it “OracleCDHToSiebelAsyncCustomer”. Drag and drop a Partner Link activity “OracleBES” and select the Adapter Wizard. Choose the”AQ Adapter” option and follow the set of steps shown in the below figure. The Service Name shown here is “Listen”.

Create a new connection to the underlying Oracle Applications instance. Specify the connection parameters and test it. This specifies the design-time connection. The run-time connection is specified in the oc4j-ra.xml and the JNDI name “eis/AQ/la2037-apps” is specified as shown in the figure below.
Select the “Dequeue” option as the AQ Adapter is a message consumer. The “APPLSYS” and “WF_DEFERRED” refer to the Oracle Applications database schema and the BES event queue.
The Consumer name in this example is set to “WF_DEFERRED” and it specifies the durable subscription id under which the adapter is registered.

The BES event payload (AQ message payload) is often empty and the hence native format translation is set to opaque. The event header is part of the AQ message headers and the next few steps will explain how to extract the Adapter message header information. Select finish. The Adapter WSDL – “Listen.wsdl” gets generated and the WSDL extension contains the JCA artifacts.
Configuring the BPEL receive activity to consume the BES event

Drag & drop the BPEL receive activity and point to the Partner Link. Select the desired operation. The only operation that is available in this example is “Dequeue”. Select “createInstance” as the incoming event is used to trigger the BPEL process.

To extract the Adapter headers, go to the Adapters tab of the Receive activity. Select the browse button.

This launches the “Variable Chooser” window. Select the “Create” option. Select “Variables” folder under “Process”. The variable created for this example is “PayloadHeader”.

Choose Message Type as the option and point it to the Adapter WSDL – Header_msg type.

The structure of the “PayloadHeader” is shown in the below figure.
Configuring DB Adapter Stored Procedure Invocation

The next set of steps show the configuration of DB Adapter Stored Procedure invocation to fetch the entire payload associated with the BES event from Oracle Applications. Drag & drop a Parter Link and name it Oracle CDH. Use the Adapter Wizard (3rd option) to create the adapter configuration. Choose the “DB Adapter” option and name the adapter service as “getCustomer”.

![Variable Chooser](Image)

![Adapter Configuration Wizard](Image)
Specify the connection information of the database adapter. The JNDI name points to the runtime connection information and must be set in the oc4j-ra.xml file.

**Image Description:**
- **Connection Information:**
  - Connection: la2037
  - User Name: apps
  - Driver: oracle.jdbc.driver.OracleDriver

- **Database Server JNDI Name:** ois/D8/la2037-apps

**Adapter Configuration Wizard - Step 3 of 3: Specify Stored Procedure**

- **Schema:** apps
- **Procedure:** XXBPEL_PARTY_FUB.GET_PARTY
- **Arguments Table:**
  - P_PARTY_ID | NUMERIC | IN | 1
  - P_PARTY_OBJECT | XXBPEL_PARTY_FUB | OUT | 2
  - P_RETURN_STATUS | VARCHAR2 | OUT | 3
  - P_RETURN_MSG | VARCHAR2 | OUT | 4
Select the “APPS” schema and the desired Stored Procedure. The “XXBPEL_PARTY_PUB.GET_PARTY” is a custom stored procedure and is not packaged with the example. Select Next and then Finish to generate the Adapter WSDL.

Configure a BPEL invoke activity for the DB Adapter Stored procedure invocation
Drag & drop an “invoke” activity and point it to the “OracleCDH” Partner Link. Select “getCustomer” as the operation and automatically generate input and output variables.

**Use transform activity to create the DB Adapter input message**

Drag & drop a “transform” activity and name it “Extract Customer Key”. This transform activity is used to extract the key fields of the BES event from the AQ message header and set it to the DB Adapter Stored Procedure invoke request.

The “OracleCDH” Partner Link takes the Party ID as input. The incoming BES AQ message header has a standard AQ Header and in addition contains the AQ Payload Header as well. The AQ Payload Header contains a Parameter List which consists of Name/Value pairs one of which is Party ID. Select the “for-each” function under “XSLT constructs” and drag & drop it to Party-ID on the right-hand side. The following XPATH expression is specified to iterate through the Parameter List and select the Party ID parameter. The XPATH expression is given below:

```
/tns:Header/tns:PayloadHeader/PARAMETER_LIST/PARAMETER_LIST_ITEM[NAME = "PARTY_ID"]
```
Configuring Partner Link to invoke Siebel Service

Drag & drop a Partner Link activity and browse (1st option) the Siebel Adapter WSDL generated earlier. Configure as shown in figure below.
Configuring invoke activity for the Siebel Adapter Service

Select the “insert” operation and autogenerate the input and output variables. Configure as shown in the above figure.
Configure transform activity to generate Siebel Adapter input

Drag & drop a transform activity. Point the DB Adapter Stored Procedure invoke output to the Siebel Adapter input as shown in the above figure. Select the edit button to specify the transformations.

Set the location on the RHS to "S/BO/Contact/Account/Insert" string as shown in the below figure.

Map City, Country, State, County and other fields as shown in the below figure.
The XSL for the above transformation is shown in the figure below.

```xml
  <xml:template match="/">
    <xml:insert>
      <xml:attribute name="location">
        <xml:value-of select="/3/B/Contact/Account/insert"/>
      </xml:attribute>
      <xml:insert>
        <xml:attribute name="City">
        </xml:attribute>
        <xml:attribute name="Country">
        </xml:attribute>
        <xml:attribute name="County">
        </xml:attribute>
        <xml:attribute name="State">
        </xml:attribute>
        <xml:attribute name="Street">
        </xml:attribute>
        <xml:attribute name="Street">
        </xml:attribute>
        <xml:attribute name="Street">
        </xml:attribute>
      </xml:insert>
  </xml:template>
</xml:stylesheet>
```

The complete process is shown in the figure below.
Deploy the BPEL process.

**Testing the BPEL process**

Go to the Oracle Applications (E-Business Suite) Home Page and log in.

Navigate to Customer page. Select Organization and then “Create Organization” tab on that page. Specify the parameters for creating an Organization object and then select “Apply”.
This will trigger the BES event. Please refer to Oracle Applications documentation for configuring of the BES. The state of the process instance can be monitored from the BPEL Console, the management and monitoring tool of the BPEL Process Manager. Navigate to the “Flow” and the “Audit” pages for further details. A successful run of the example scenario is captured in the next few screens.
New instance of BPEL process "OracleCDHToSiebelAsyncCustomer" initiated (# "9701").

[2006/05/27 19:44:10] Received "ListenToBES_Dequeue" call from partner "OracleBES" less

[2006/05/27 19:44:10] Updated variable "GetCustomer_getCustomer_inputVariable" less

[2006/05/27 19:44:10] Invoked 2-way operation "getCustomer" on partner "OracleCDH" less
[2005/02/27 13:44:00] Updated variable "invoke_Siebel_insert_inputVariable" less

<Invoke_Siebel_insert_InputVariable>
    <ml:insert>
      <ml:City>New York</ml:City>
      <ml:Name>test121</ml:Name>
      <ml:Street_spcAddress>222, Oracle</ml:Street_spcAddress>
    </ml:insert>
  </Siebel>
</Invoke_Siebel_insert_InputVariable>

[2005/02/27 13:44:57] Invoked 2-way operation "insert" on partner "SiebelCustAccountInsert": less

<Invoke_Siebel_insert_InputVariable>
    <ml:insert>
      <ml:City>New York</ml:City>
      <ml:Name>test121</ml:Name>
      <ml:Street_spcAddress>222, Oracle</ml:Street_spcAddress>
    </ml:insert>
  </Siebel>
</Invoke_Siebel_insert_InputVariable>

<InsertResponse xmlns="urn:waysoftware:ibm:jul2003:insertResponse" id="1C687B8B13DF894AB15CD2B5C0F3E5"/>
</InsertResponse>
</Invoke_Siebel_insert_OutputVariable>