Oracle OpenWorld 2008

Hands on Lab

Service Enable Oracle JD Edwards EnterpriseOne with Oracle SOA Suite BPEL Process Manager

Table of Contents

1 TUTORIAL OVERVIEW 3
2 CREATING THE BPEL PROCESS 3
2.1 Startup 3
2.2 Creating the BPEL Workspace and Project 4
2.3 Creating a Partner Link to Insert the JDE Customer Credit Query 8
2.4 Creating Invoke Activity 11
2.5 Creating an Initial Assign to Send Input Request Data to the Invoke 15
2.6 Creating an Assign to Send Output Data from the Invoke 17
2.7 Validating, Compiling and Deploying the BPEL Process 19
2.8 Example of BPEL Process 20
3 TEST THE INTEGRATION 21
3.1 Testing the BPEL Process from the BPEL Console 21
4 ADDITIONAL CREDIT 24
4.1 BPEL Console 24
4.2 JDeveloper 25

1 Lab Overview

This lab demonstrates native web services integration to JDEdwards EnterpriseOne using Oracle BPEL Process Manager. JDEdwards EnterpriseOne release 8.97 has enhanced support for native web services, and exposes a number of functions as Business Services.

In this tutorial you will Query Customer Credit Information from JD Edwards EnterpriseOne using using and Oracle BPEL Process Manager.

In this lab

- Generate a WSDL in EnterpriseOne for Customer Credit information*
- Consume the WSDL in Oracle BPEL Process Manager as a Partnerlink
- Invoke the getCustomerCreditInfo() operation within the WSDL
- Pass in ID as a parameter and the expected response will be credit details.

*We have completed the first step for you, and the generated WSDL is located in your desktop folder FMW4Apps → LabFiles → JDE. This was done primarily to ensure we finish the lab in the short duration.

Software Used:

- JDEdwards EnterpriseOne Apps 8.12 with Tools 8.97
- Oracle SOA Suite version 10.1.3.1 [available for download from OTN]
- Oracle JDeveloper version 10.1.3.4 [available for download from OTN]

2 Creating the BPEL Process

2.1 Startup

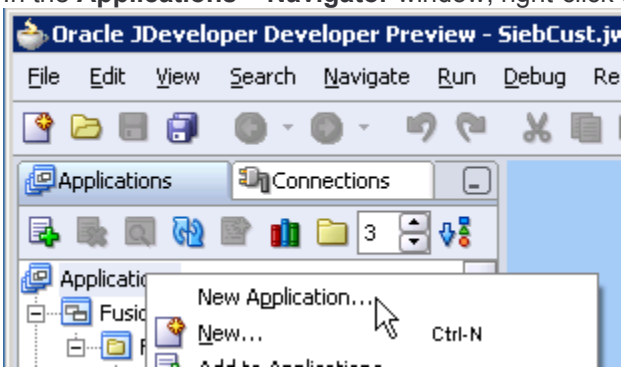
Start **Oracle SOA Suite** by clicking the StartBPEL shortcut in FMW4Apps → Shortcuts Folder on your desktop.

When it completes you will see a check mark like this:

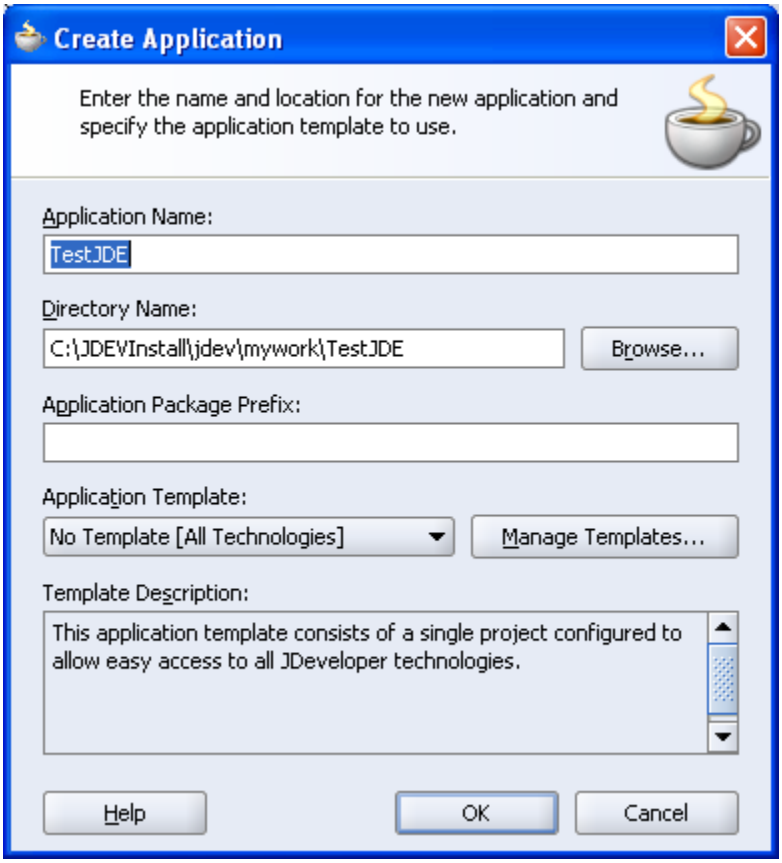
Then start **Oracle Jdeveloper** by clicking on the icon as below in the same folder.

2.2 Creating the BPEL Workspace and Project

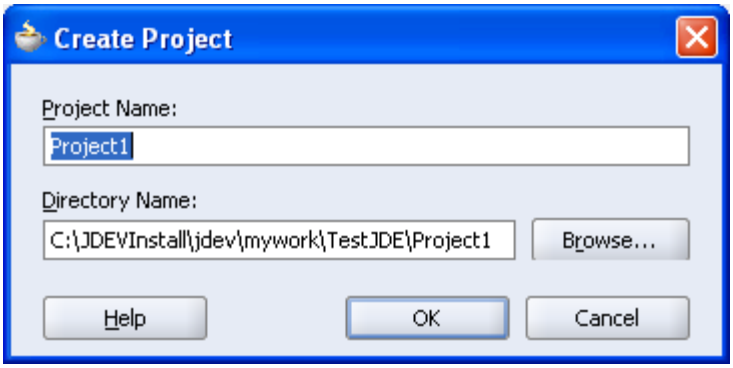
In the **Applications – Navigator** window, right-click on **Applications** and select **New Application…**



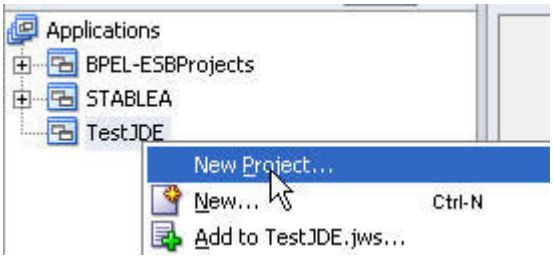
Name your application. In this example, we'll use **TestJDE**. Your directory names may be different. Take the defaults as-is. Click OK to create the application.



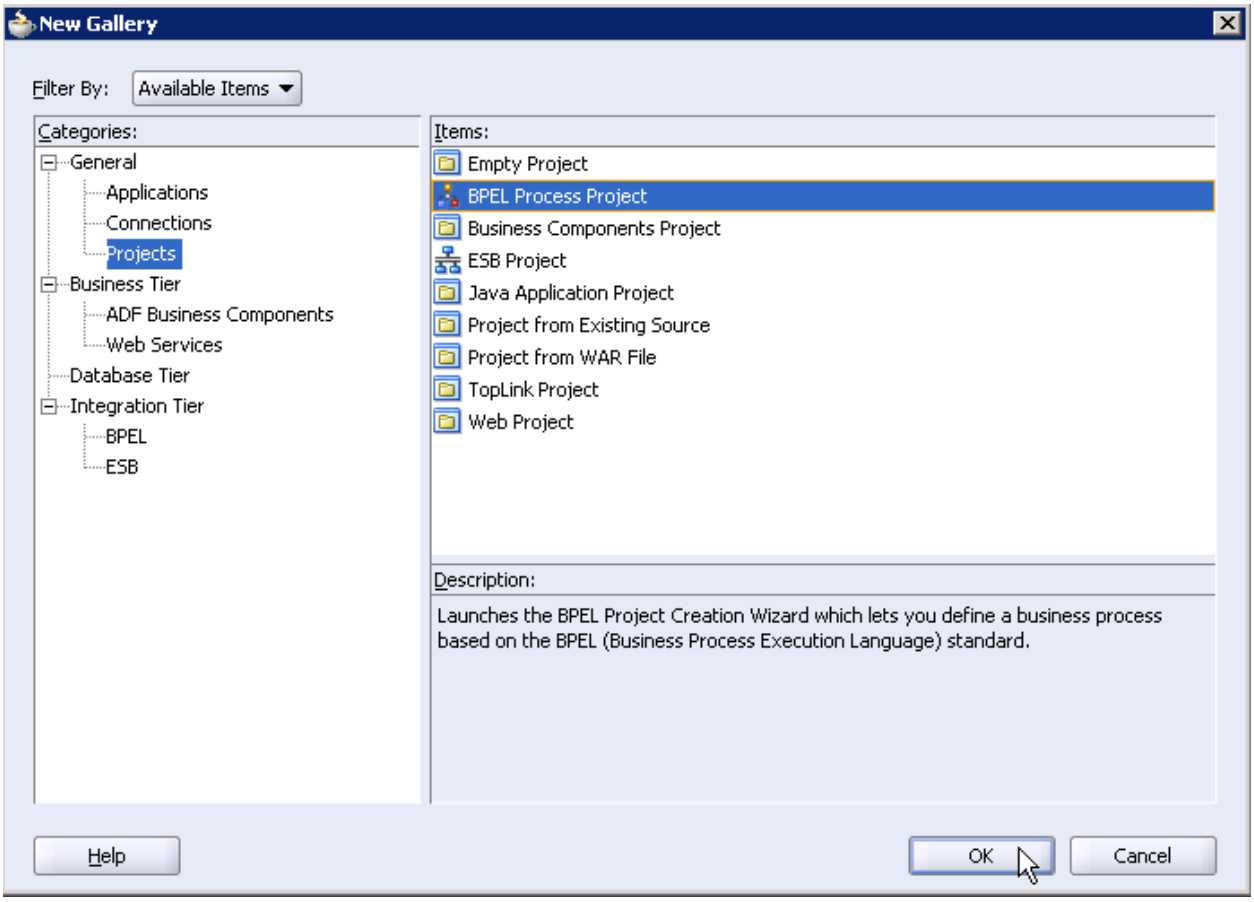
A Create Project window will appear. Click Cancel. (Yes Click Cancel).



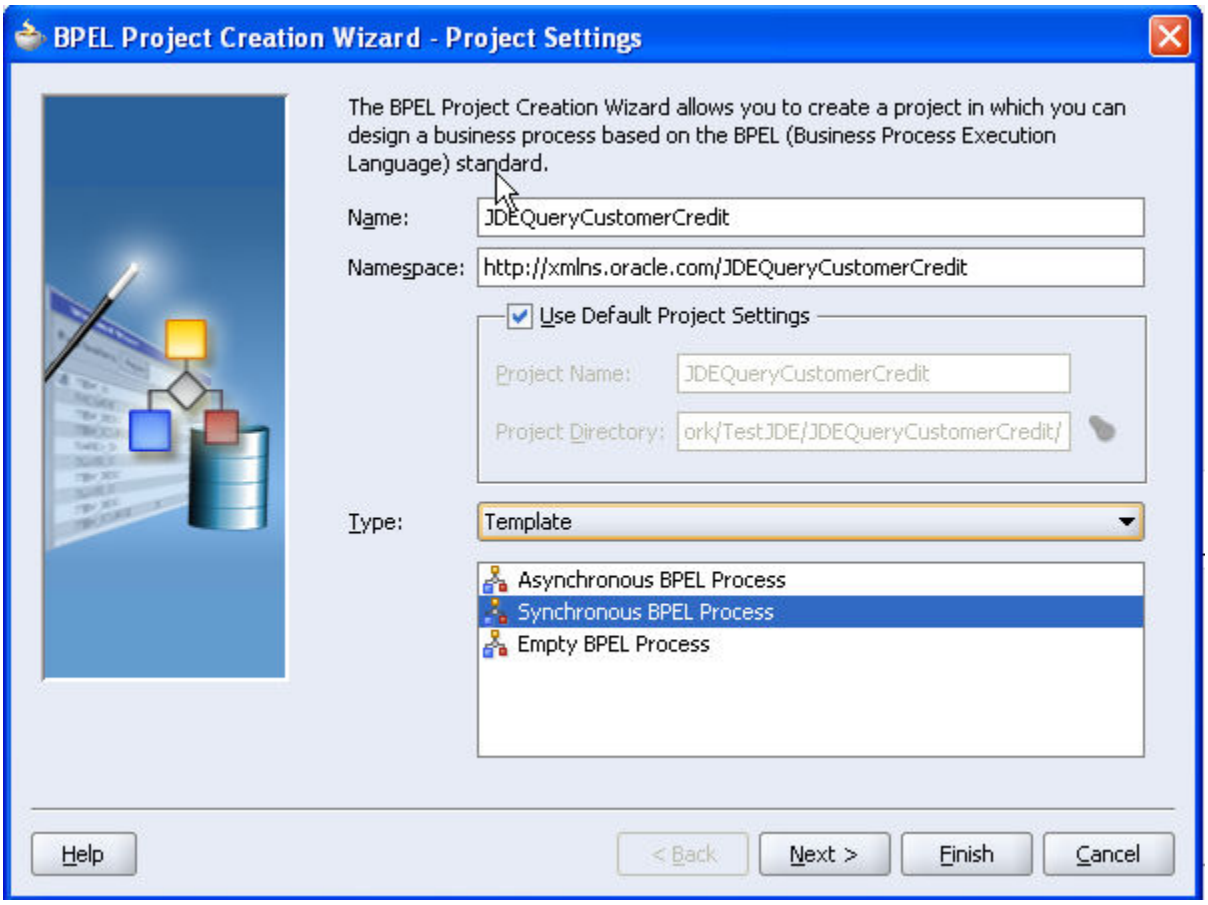
You will now be back at the main JDeveloper window, and you will see your **TestJDE** in the applications window. Right click on the TestJDE application, and select New Project…



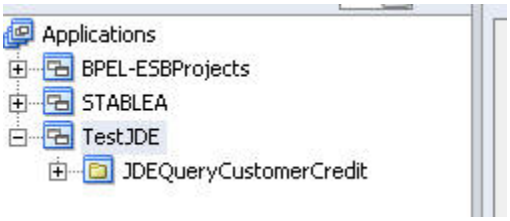
In the New Gallery window, select BPEL Process Project and click OK.



Name the process **JDEQueryCustomerCredit** in the Name field and select **Synchronous BPEL Process** as the template. Click Finish.

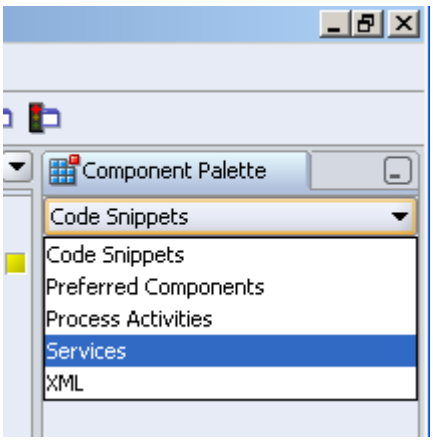


The JDEQueryCustomerCredit project is then created under your TestJDE application. You can see the files in the **Applications – Navigator** window.

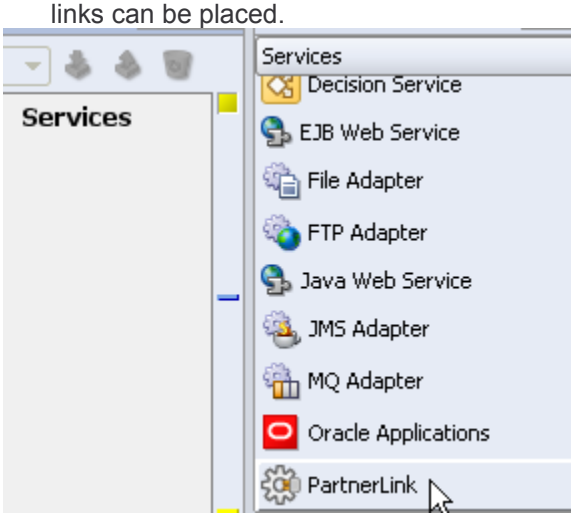


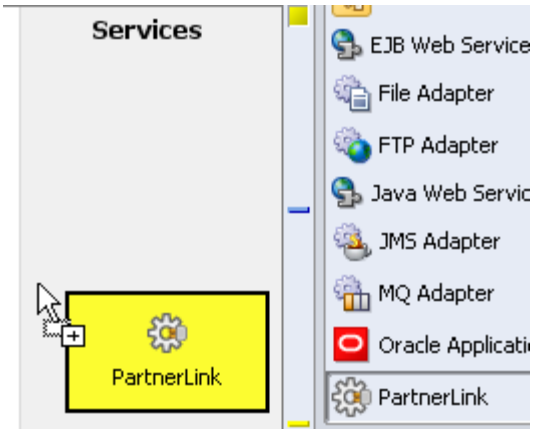
2.3 Creating a Partner Link to Insert the JDE Customer Credit Query

1. Select **Services** from the drop-down list of the **Component Palette** section in the upper right section of the JDeveloper Designer.

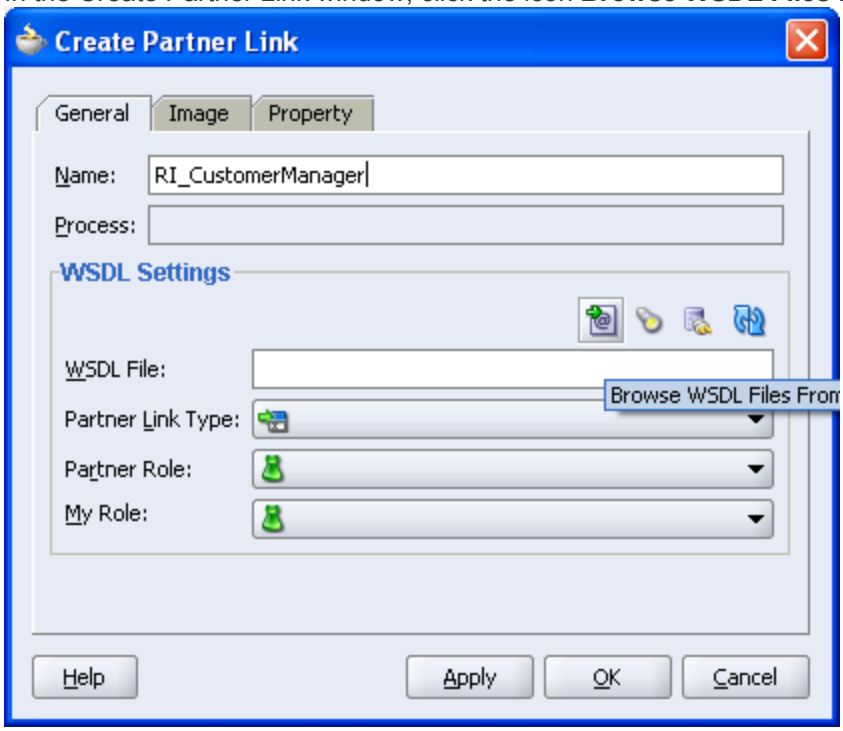


2. Drag and drop a **PartnerLink** activity onto the **right** side of the designer window anywhere beneath the Services header. Note that both sides of the designer window have sections in which partner links can be placed.

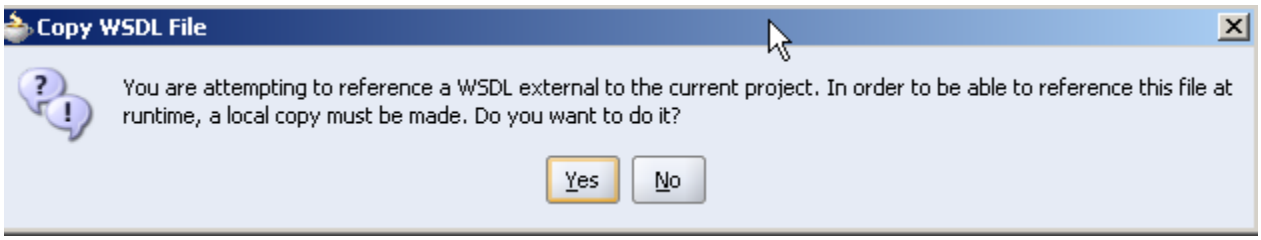




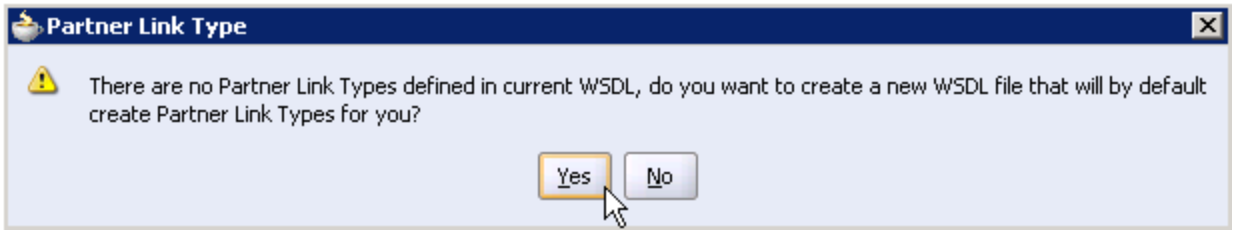
3. In the Create Partner Link window, click the icon **Browse WSDL Files From Local File System.**



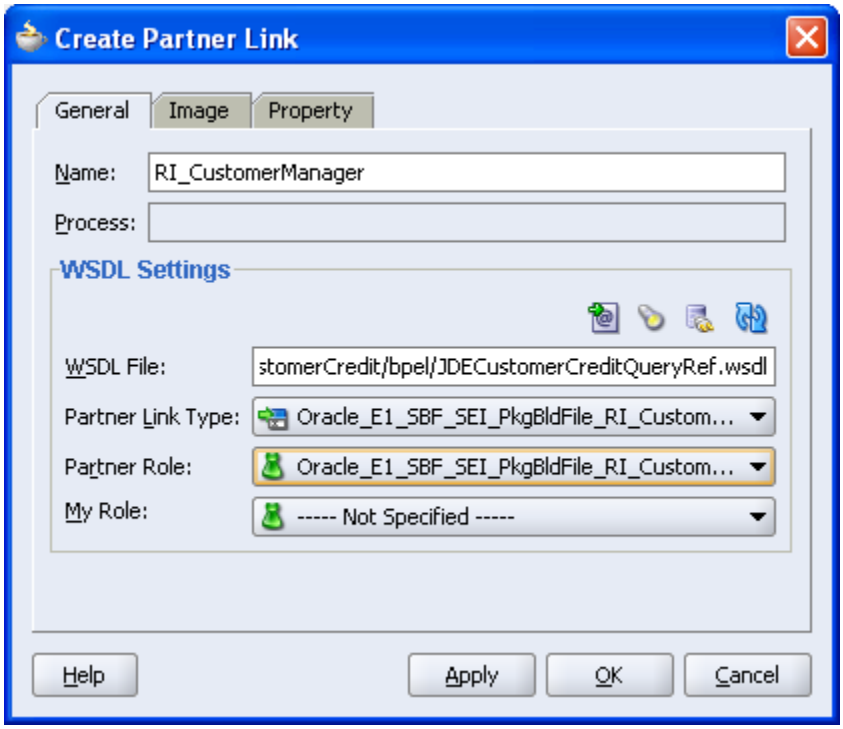
4. Browse and select the JDE WSDL file JDECustomerCreditQuery.wsdl. It will located in your desktop folder FMW4Apps → LabFiles → JDE



You will get a Partner Link Type box, select Yes.

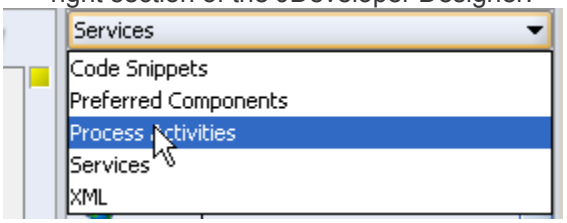


Enter the following values to finish creating the partner link
Name: **RI_CustomerManager**
Partner Role: **Oracle_E1_SBF_SEI_PkgBldFile_RI_CustomerManager_Role**
Partner Link Type is automatically selected.
Click **Ok**

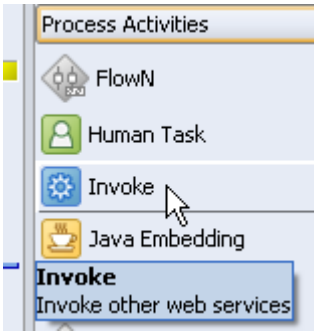


2.4 Creating Invoke Activity

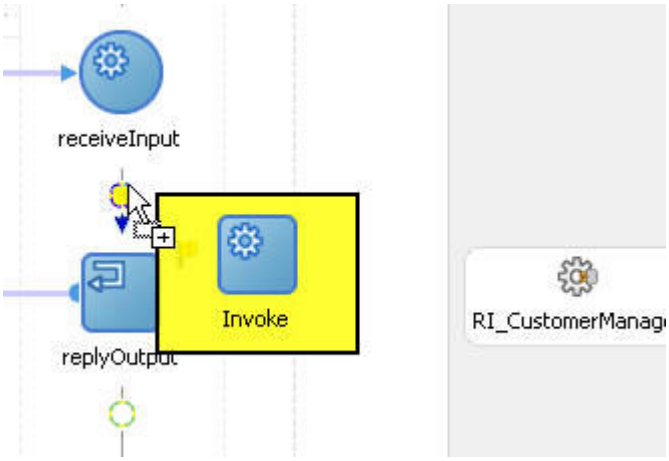
1. Select **Process Activities** from the drop-down list of the **Component Palette** section in the upper right section of the JDeveloper Designer.

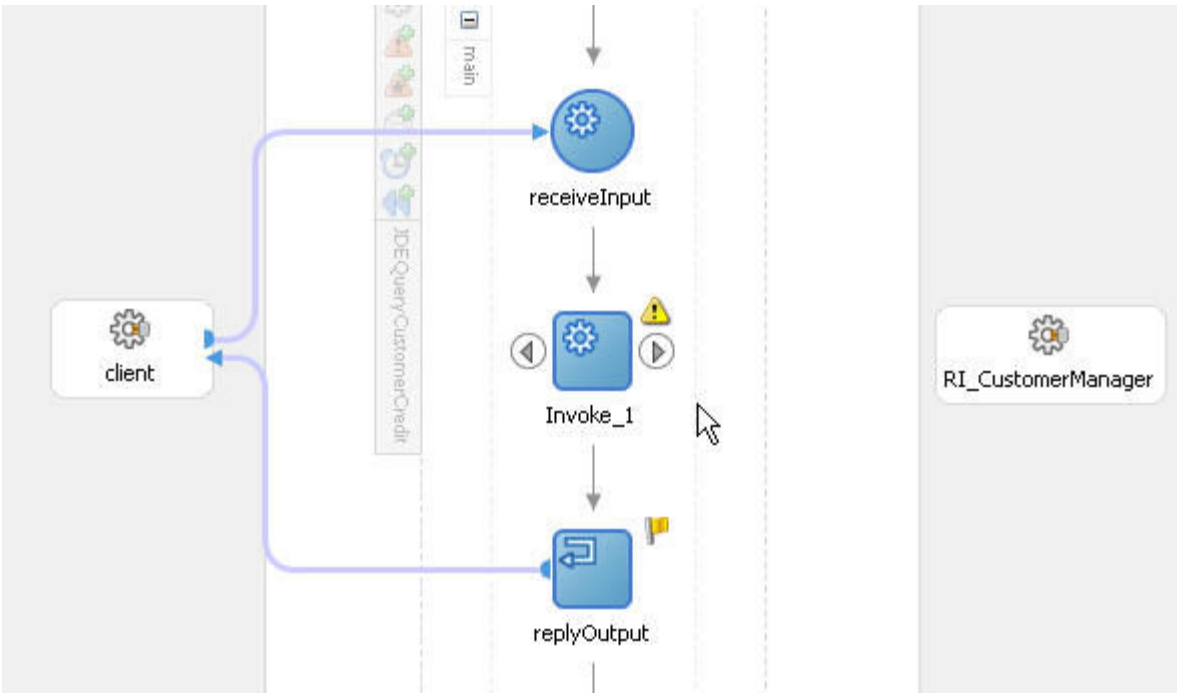


2. Drag and drop an **Invoke** from the Component Palette section.



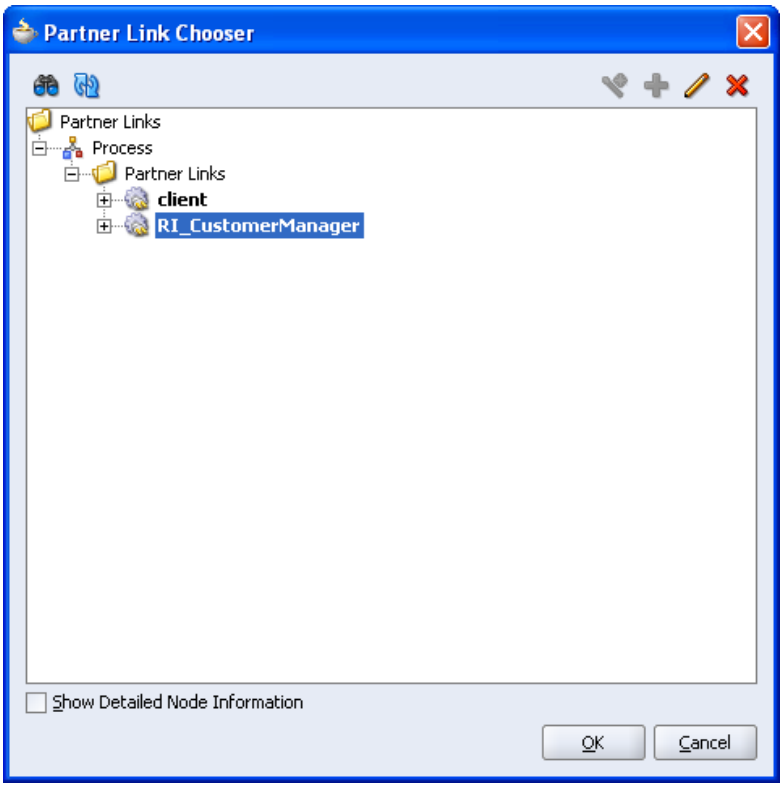
And drop onto the canvas immediately **after** the **receiveInput** activity



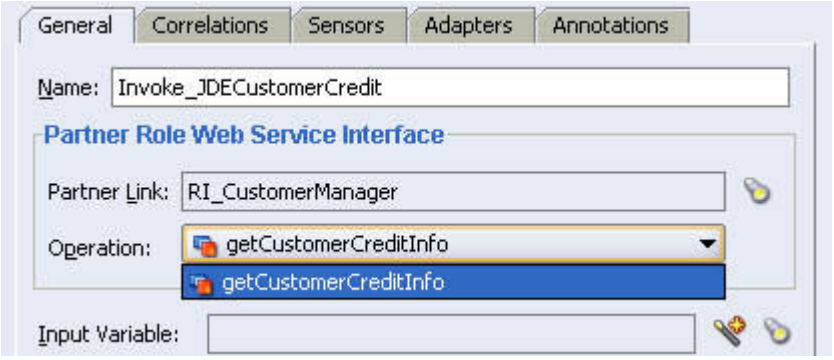


3. Double-click the **Invoke_1** icon to display the Invoke window

4. Enter the following values.
    a. Set Invoke Name to **Invoke_JDECustomerCredit**
    b. Set Partner Link to **RI_CustomerManager** by selecting the flashlight icon clicking OK.

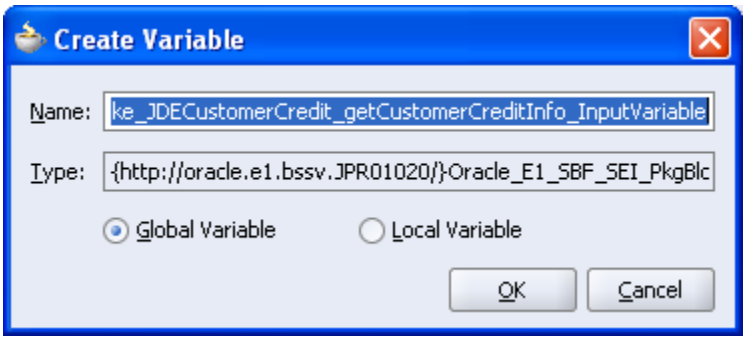


    c. Set the Operation to **getCustomerCreditInfo**



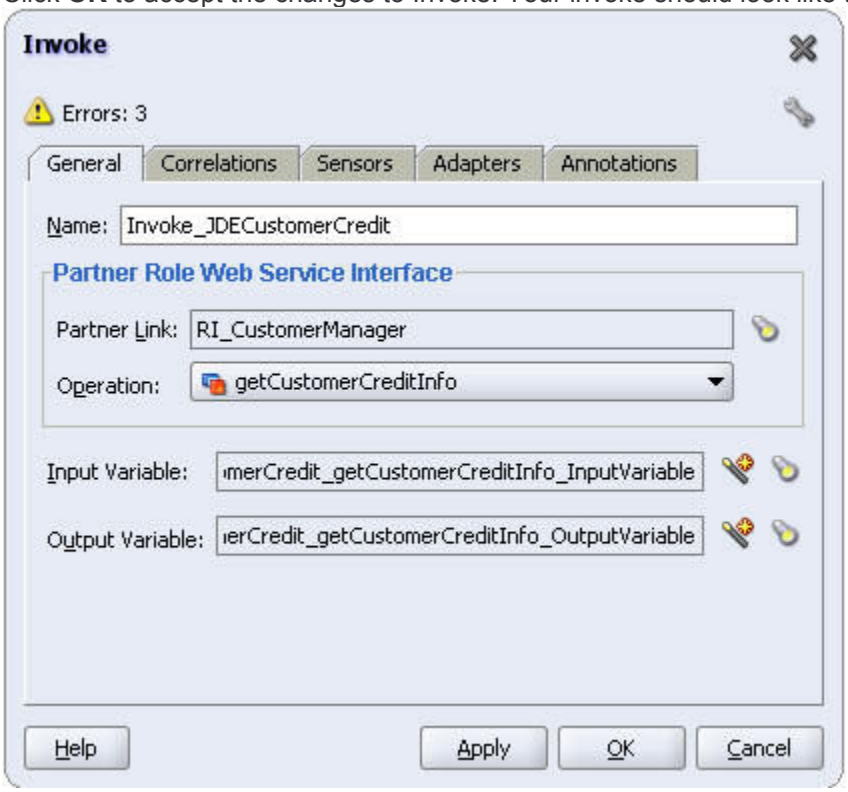
    d. Click on the '**Auto-create variable**' magic wand icon next to the **Input Variable** text field to create a variable for this activity.

Click **OK** to accept the default variable name.



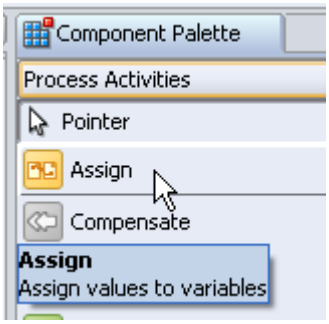
e. Click on the '**Auto-create variable**' magic wand icon next to the **Output Variable** text field to create a variable for this activity. Click **OK** to accept the default variable name, as done for previous step.

f. Click **OK** to accept the changes to Invoke. Your invoke should look like this.

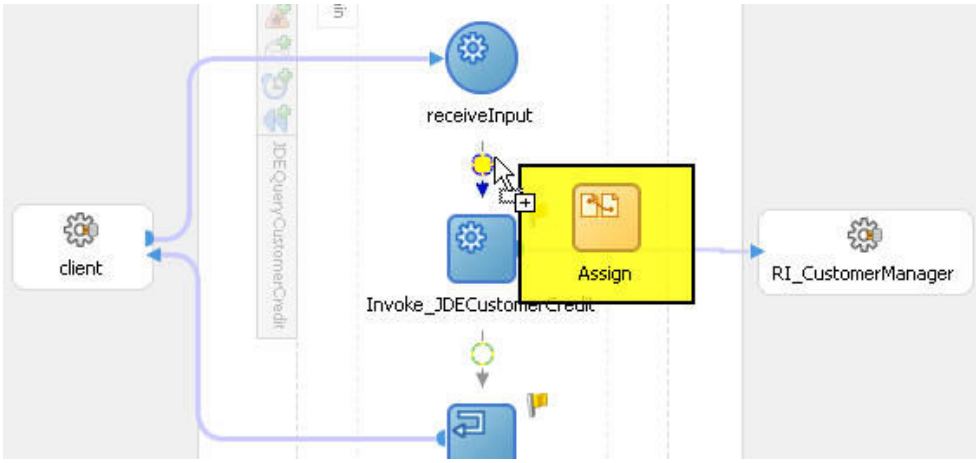


2.5 Creating an Initial Assign to Send Input Request Data to the Invoke

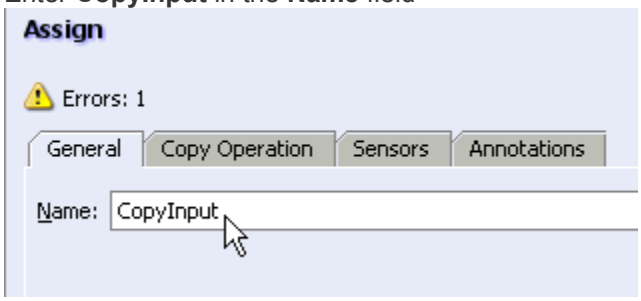
1. Verify **Process Activities** is still selected from the drop-down list of the **Component Palette** section in the upper right section of the JDeveloper Designer.



2. Drag and drop an **Assign** activity from the Component Palette section between the **receiveInput** and **Invoke_JDECustomerCredit** activities so we can copy the **input** field in the input Variable to the Invoke input variable.
3.

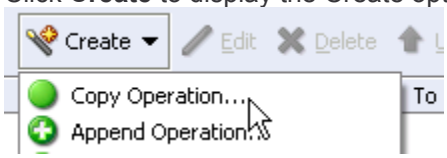


4. Double-click the **Assign** icon that you just dropped to display the Assign window.
5. Click the **General** tab
6. Enter **CopyInput** in the **Name** field

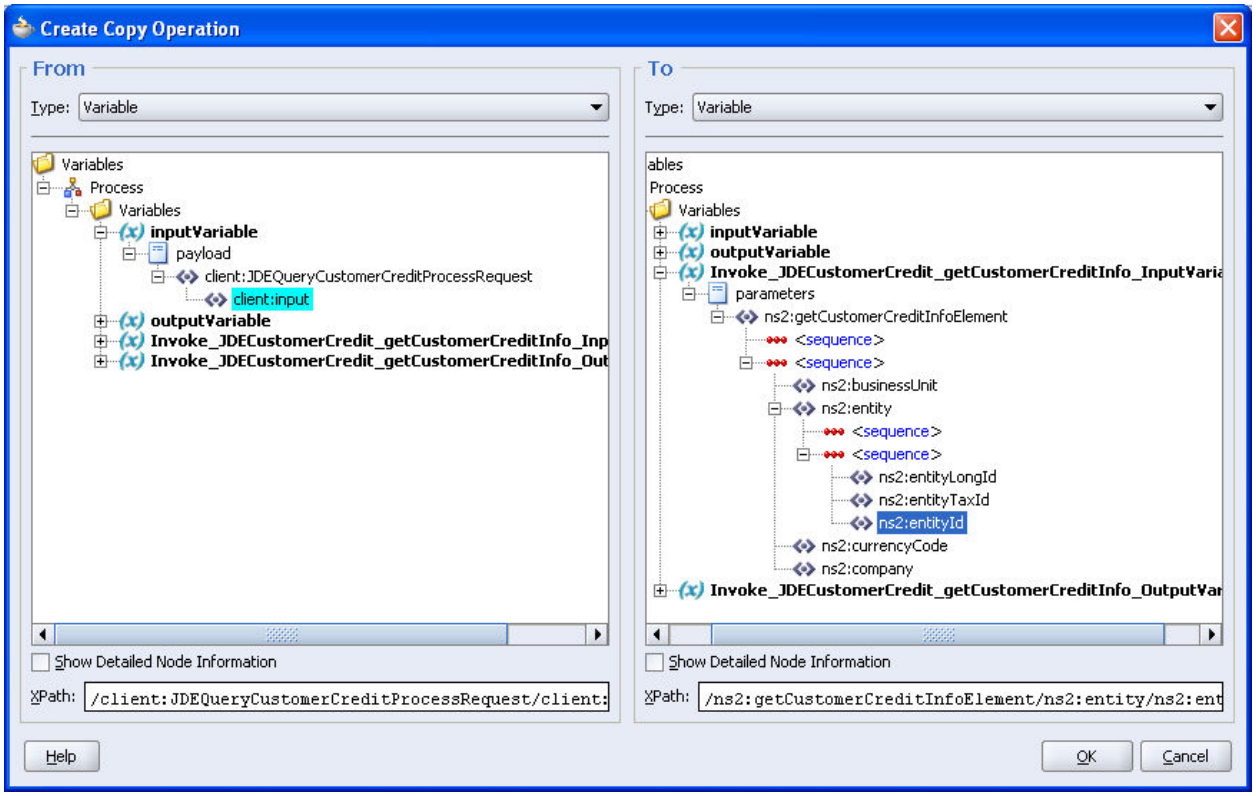


7. Click **Apply**
8. Click the **Copy Operation** tab

9. Click **Create** to display the Create options window, and then select Copy Operation…

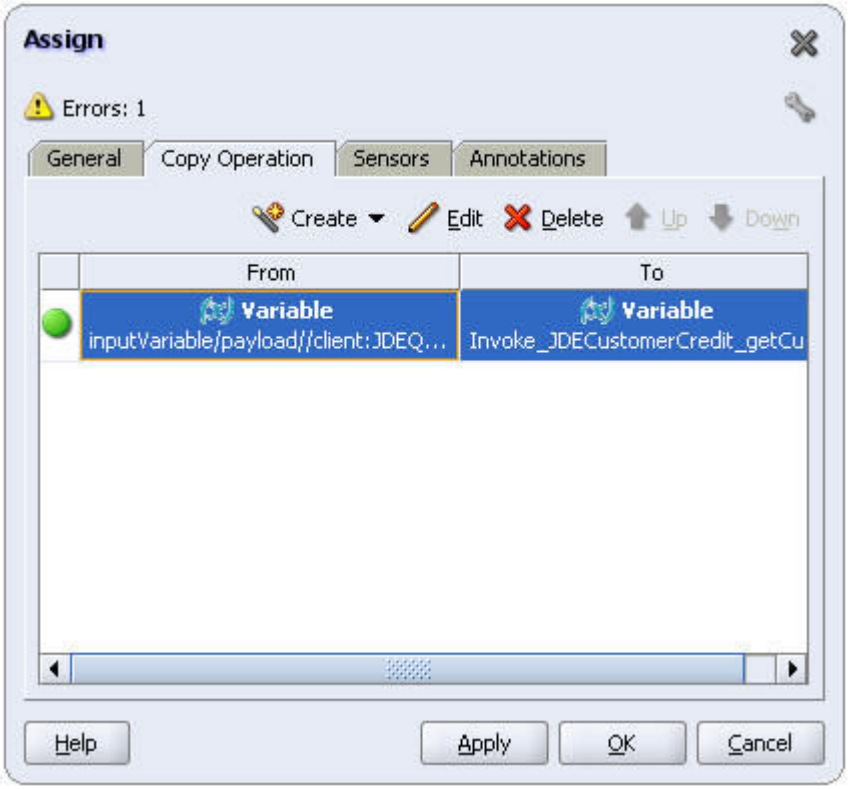


10. Make the changes to appear as follows:



Note that we are taking the user input (client:input variable) and assigning it to the input variable (entityId: variable that represents the E1 address book number) of the Invoke step for the JDE web service.

11. Click **OK** to close the Create Copy Rule window and the Assign window

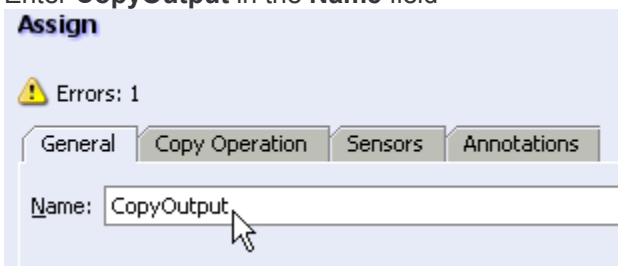


2.6 Creating an Assign to Send Output Data from the Invoke

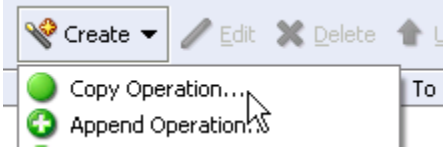
Now, we will repeat the steps above to set the response from JDE Query call to the output variable. This time the Assign will be **AFTER** the **Invoke.**

1. Verify **Process Activities** is still selected from the drop-down list of the **Component Palette** section in the upper right section of the JDeveloper Designer.
2. Drag and drop an **Assign** activity from the Component Palette section between the **Invoke_JDECustomerCredit** and **replyOutput** activities so we can copy the **output** field in the Invoke output Variable to the replyOutput input variable.
3. Double-click the **Assign** icon that you just dropped to display the Assign window.

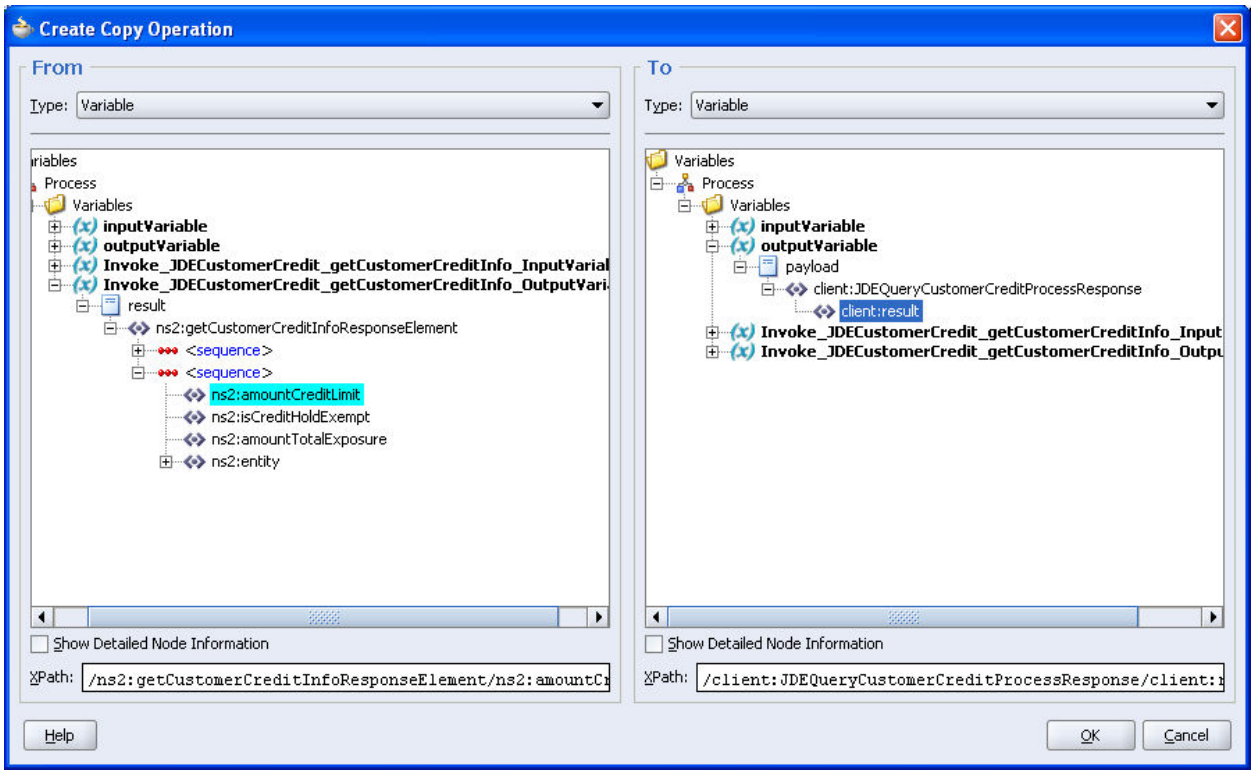
4. Click the **General** tab
5. Enter **CopyOutput** in the **Name** field



6. Click **Apply**
7. Click the **Copy Operation** tab
8. Click **Create** to display the Create options window, and then select Copy Operation…



9. Make the changes to appear as follows:



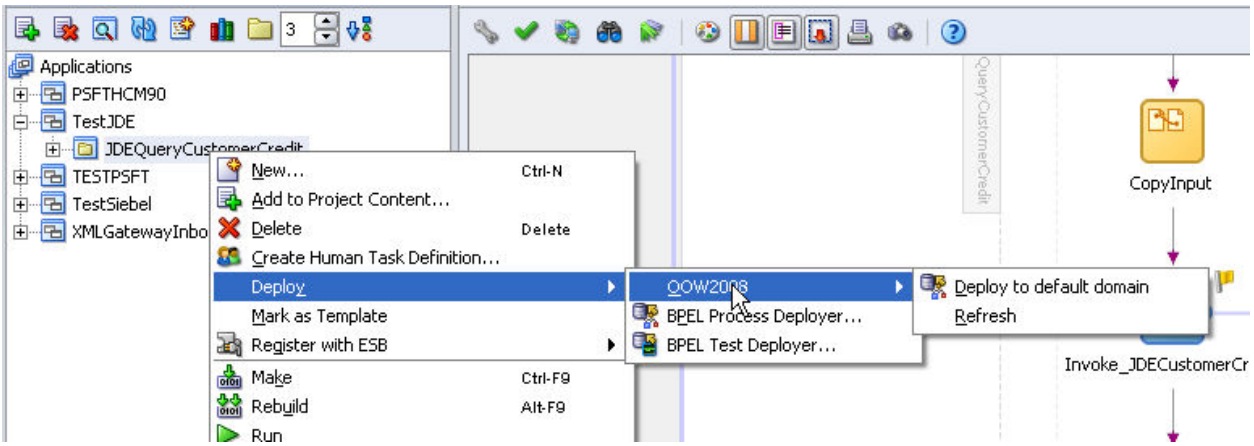
Note here that we are taking the output of the Invoke step (amountCreditLimit) and assigning it to output variable (amountCreditLimit) of the bpel process. This is result that user will see returned in BPEL console.

10. Click **OK** to close the Create Copy Rule window and the Assign window. At this point you are done building your BPEL process. Now lets build and deploy this.

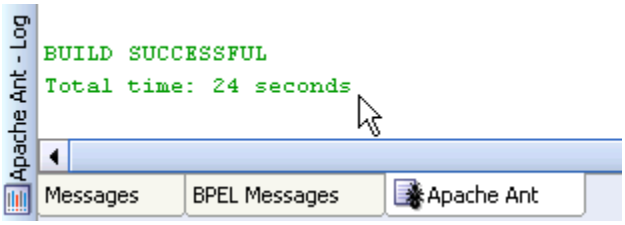
2.7 Validating, Compiling and Deploying the BPEL Process

Note: For this section you need to ensure you have BPEL manager started and a link to the default domain from Jdeveloper.

1. Go to the **Applications Navigator** section
2. Right-click the **JDEQueryCustomerCredit** BPEL Project
3. Select **Deploy → OOW2008→ Deploy to default domain**

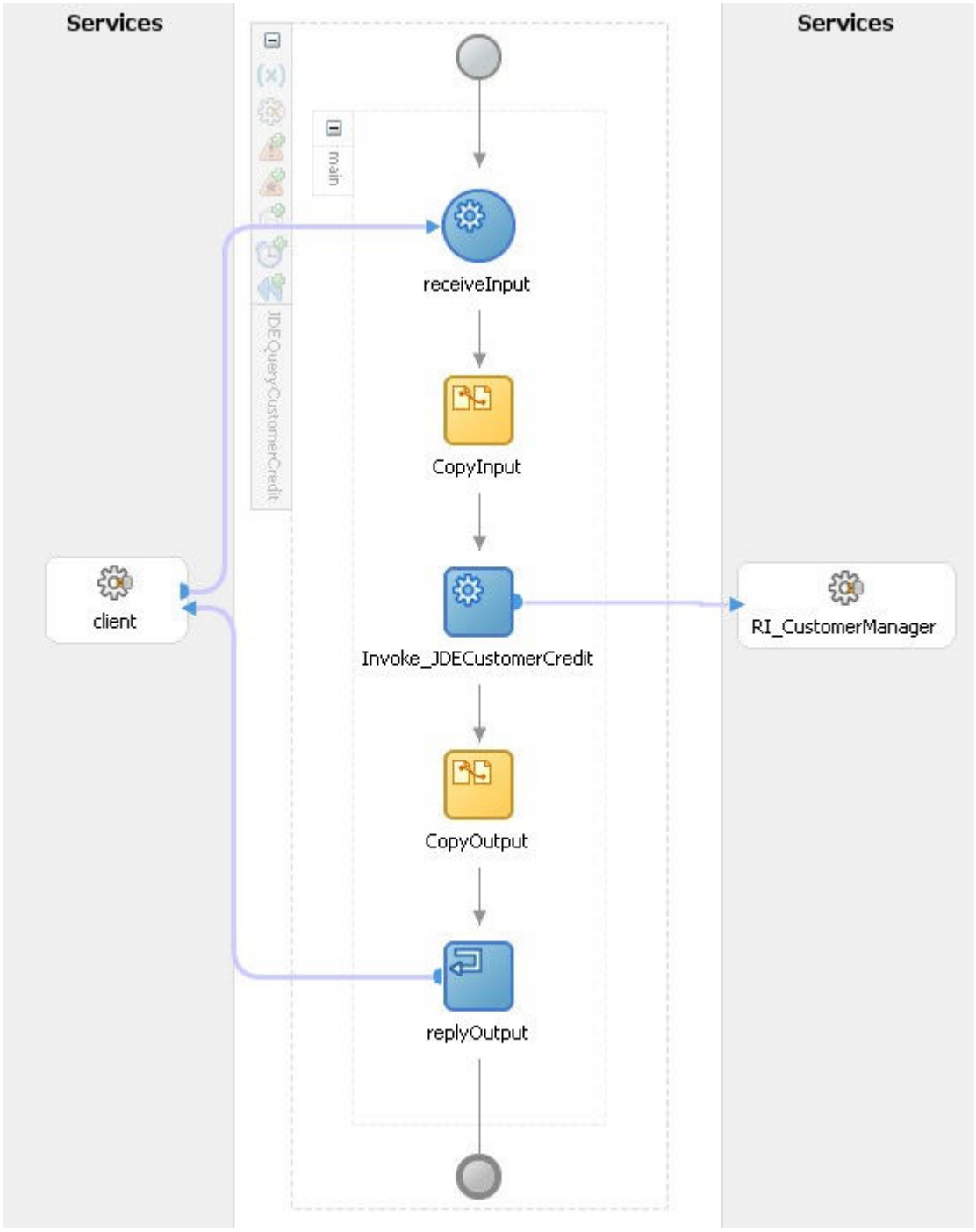


4. This compiles and deploys the BPEL process. Review the bottom window for any errors. If there are no errors, deployment was successful.



5. If you have errors, make appropriate corrections and deploy again. Ignore any warnings.
6. Once you deploy with no errors, your workflow project is now ready for testing.

2.8 Example of BPEL Process



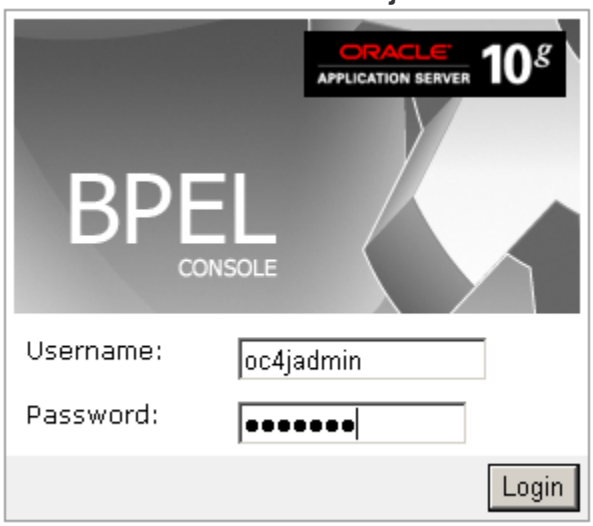
3 Test The Integration

3.1 Testing the BPEL Process from the BPEL Console

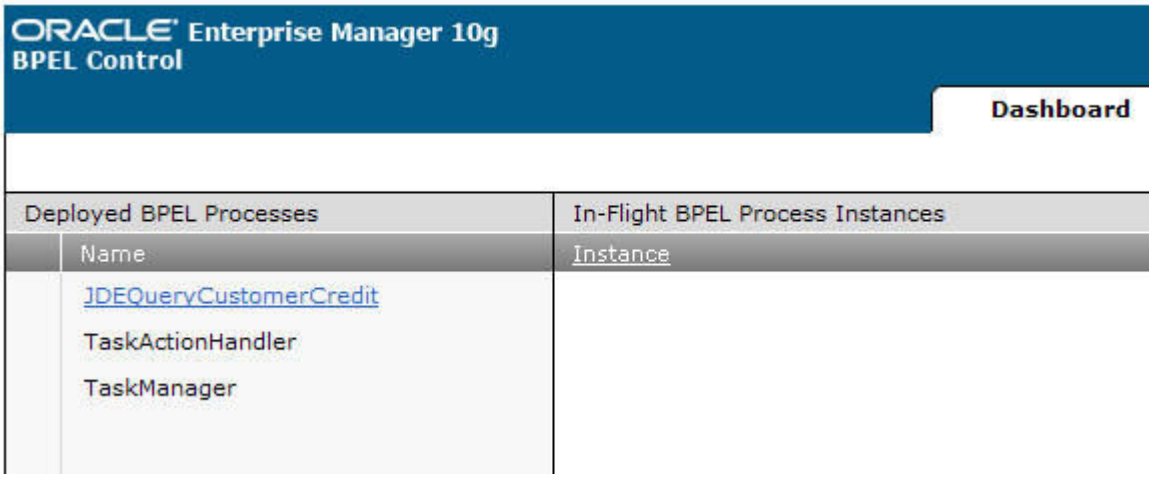
You can test that your BPEL project works by initiating an Insert from the BPEL Console.

Log into the Oracle BPEL Console using Internet Explorer by clicking on the URL icon in your FMW4Apps → Shortcuts folder.

1. The default Username is **oc4jadmin** and the default password is **welcome1**



2. Click on **JDEQueryCustomerCredit** in the Deployed BPEL Processes list.

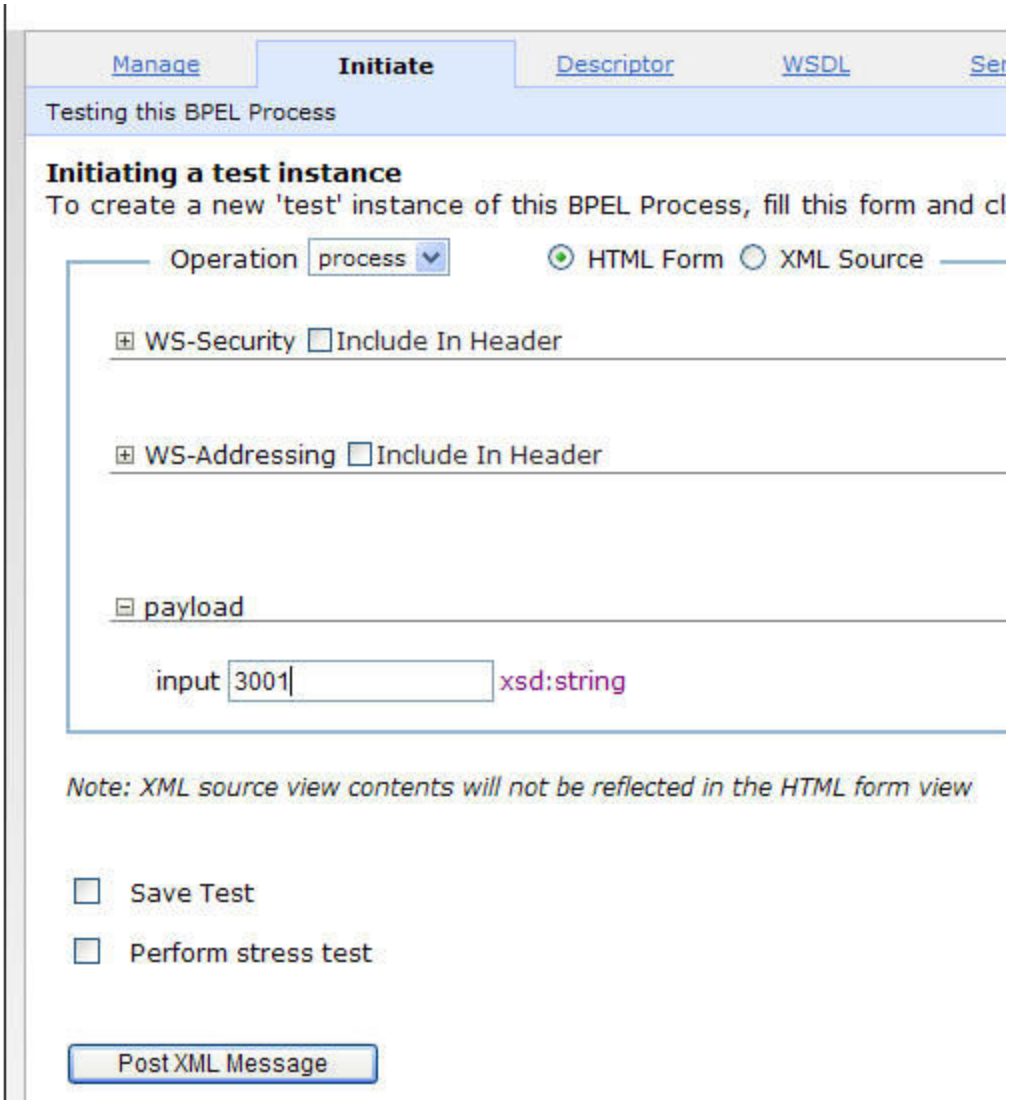


3. This takes you to the Initiate tab of the JDEQueryCustomerCredit BPEL Process where you will see a HTML form. Here enter a JDE Entity ID (Address Book Number) to Query. Sample JDE Entity Id is

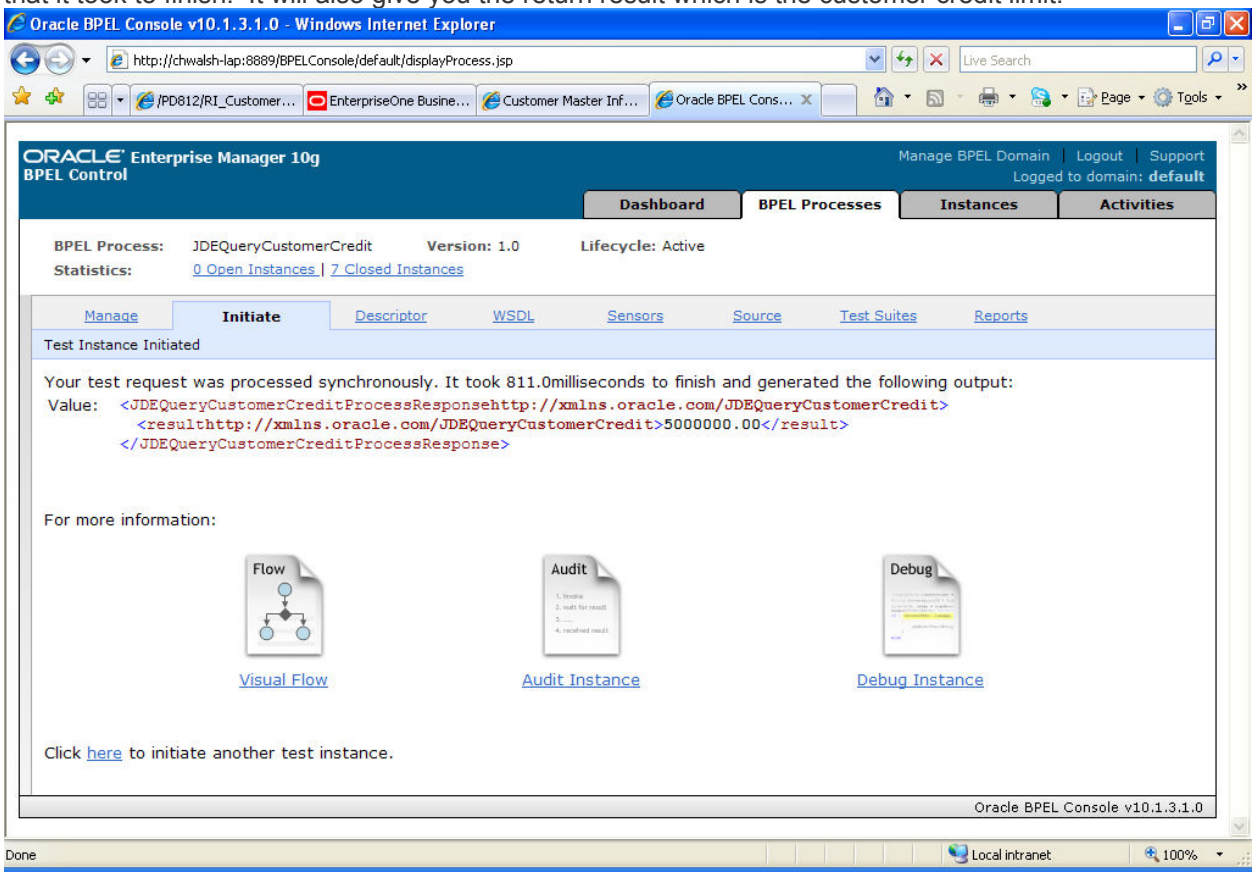
   3001

   Other samples to use are
   4242
   3111

   Enter one of these in the payload input as shown below. Then click on Post XML Message button.



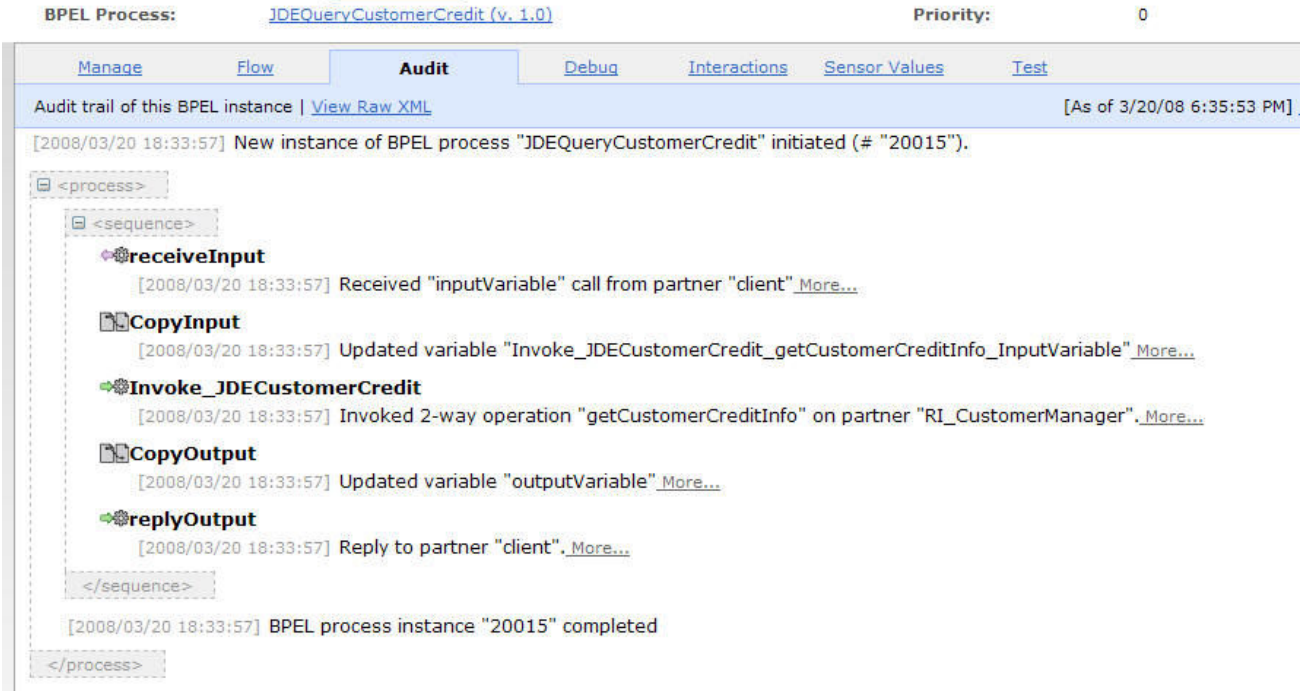
4. You should see a message saying the test request was processed synchronously, and have the time that it took to finish. It will also give you the return result which is the customer credit limit.



5. You can now click on either the Audit Instance link to verify the BPEL process completed, and look at some of the values at each step. Here is an example looking at the Audit link.

Audit Instance

A page shown below comes up. You can click on various operations in the page to see each step in the BPEL Process that queried JDE.



Click on “more” link as shown above to see additional details on process execution, such as input and output parameters.

4 Additional Credit

4.1 BPEL Console

a) Go back to Dashboard and look at completed process instances.
b) Check the executed flow diagram of completed instances. Click on each step in the process to see the XML data flowing through the step.

4.2 JDeveloper

c) Explore the different activities you can use in a BPEL process. Look at Process Activities and you will find an array of nodes to use. Interesting ones are Human Task flow, which can be used to include human approval steps and assign tasks to people. Another classic node is the Transform node, which is used to create XSLT transformations. It brings up a graphical mapper from which you can drag and drop elements.