

7 Adding Business Rules

7.1	Introduction.....	2
7.2	Description of the application.....	2
7.2.1	SOA Components	4
7.3	Prerequisites.....	4
7.4	Project Files.....	4
7.4.1	File Conventions	4
7.4.2	Installing the database schema	5
7.5	Deploying CreditCardValidation composite.....	5
7.6	Adding a Business Rule to POProcessing	7
7.7	Deploying POProcessing composite.....	14
7.8	Running the application	15
7.8.1	Additional Test Cases	16

7.1 Introduction

This tutorial shows you how to add a Business Rule to the End-2-end 105 PO Processing sample.

This tutorial can stand alone or be completed after completing chapters 1-6 of End-2-end 105 PO Processing. You can begin with the provided pre-built sample that has been completed up to Chapter 6 or, if you are building the entire application yourself, just continue here when you finish Chapter 6.

In this tutorial we begin with a SOA composite application to process and approve purchase orders. The purchase order details can come in from any source (in our case a testing page), the credit card status for the customer is validated and if the credit card is good, the order continues. An order for a large purchase price requires a manual approval step. Finally, the order is written to a text file to be processed by the fulfillment house.

This SOA composite application contains a database adapter, web services binding, Mediator ESB routing service, BPEL process, and human task. You will add a Business Rule to the composite to make the decision of whether the human task is required for manual approval or not.

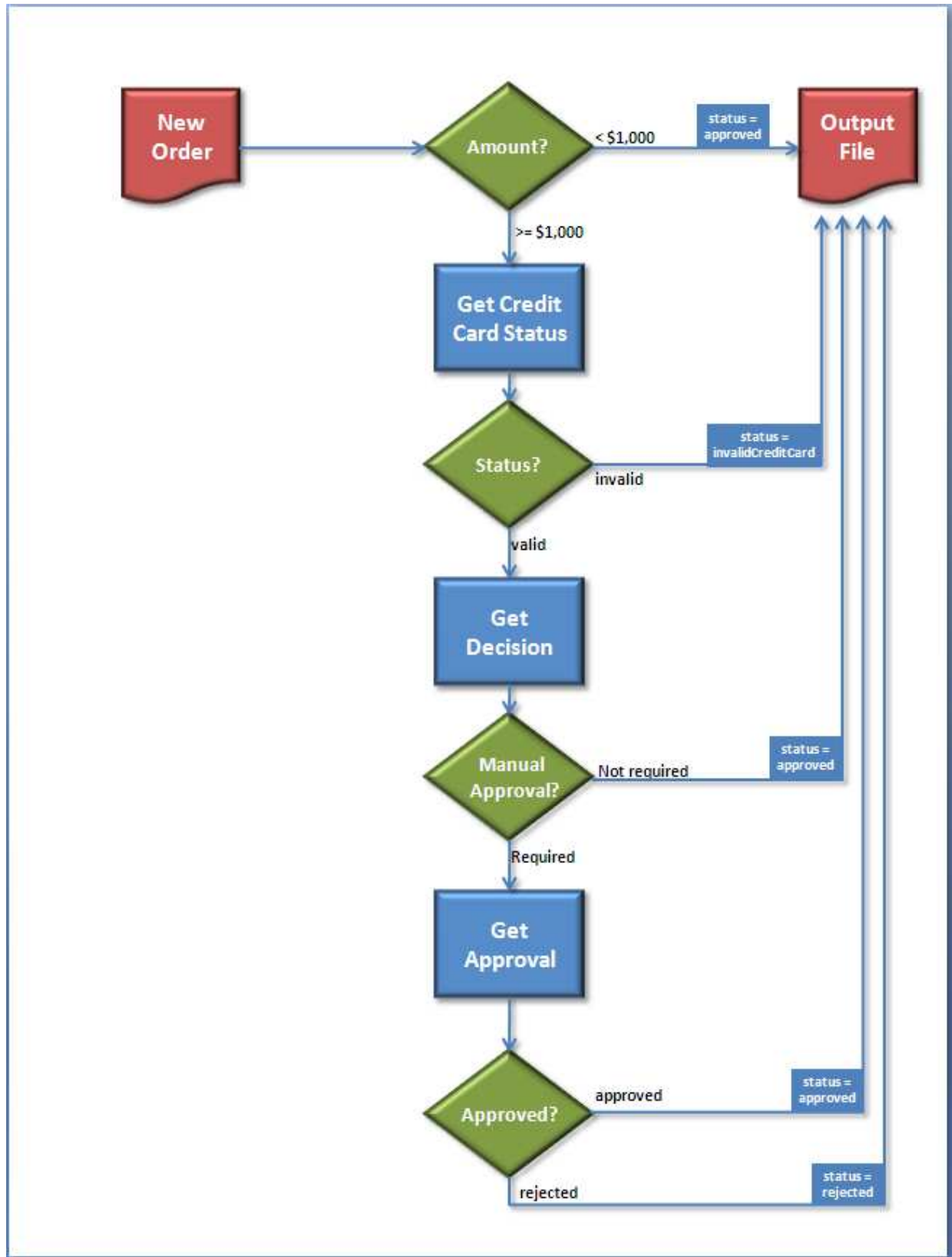
7.2 Description of the application

The application you will install is the back-end processing of a new order. The business process is as follows.

- All orders will be written to a text file to be processed by the fulfillment house.
- Small orders (defined as those under \$1,000) are to be automatically approved.
- Large orders (those greater than or equal to \$1,000) go through a validation and approval process.
 - The customer's credit card must be validated.
 - If the order is \$5,000 or more, a customer service representative must manually approve the order.
- Approved orders shall have the status "approved"
- Large orders with an invalid credit card shall have the status "invalidCreditCard".
- Large orders rejected by the customer service representative shall have the status "rejected".

Figure 1 shows a visual representation of the business process.

Figure 1 Visual view of the business process



7.2.1 SOA Components

In terms of the SOA application there are the following elements

- A service that accepts new orders in XML format.
- A File Adapter service that can write XML messages (such as orders) out to a file.
- A content-based Mediator service to route small orders to the file adapter and large orders to a BPEL process.
- A BPEL process that validates the credit card by calling a service, uses a Business Rule to check if human approval is required, and seeks human approval using a Human Task component.
- The credit card validation service used by the BPEL process to validate a given credit card.
- A Database Adapter service to retrieve the status of a given credit card from the database.

7.3 Prerequisites

This document assumes you have already installed the following:

- An Oracle database
- Oracle JDeveloper 11
- Oracle SOA Suite 11 (may be bundled with JDeveloper)

7.4 Project Files

The ZIP file for this sample came with several directories and files:

- `doc` – contains the instructions for creating and running the demo
- `input` – a few xml files with sample input data
- `sql` – two database scripts used during setup
- `schemas` – xsd files used throughout sample when defining services
- `Ch6` and `ch7` – a solution project for each chapter

7.4.1 File Conventions

After unzipping the file for this sample, move or copy it to `c:\po`. This document assumes that path. If you unzipped somewhere else then adjust accordingly when `c:\po` is referenced through this document. If you already created `c:\po` for the full tutorial or an addendum, you can safely copy this directory into it.

7.4.2 Installing the database schema

This demo requires a table in the database.

1. Create the `soademo` user. It is ok to run this script even if the `soademo` user already exists. From a command line, cd to the `c:\po\sql` directory and run the following replacing `pw` with your own system user's password:

```
cd c:\po\sql
sqlplus system/pw @create_soademo_user.sql
```

2. Create the credit card info table. It is ok to run this script more than once, even if the table already exists.

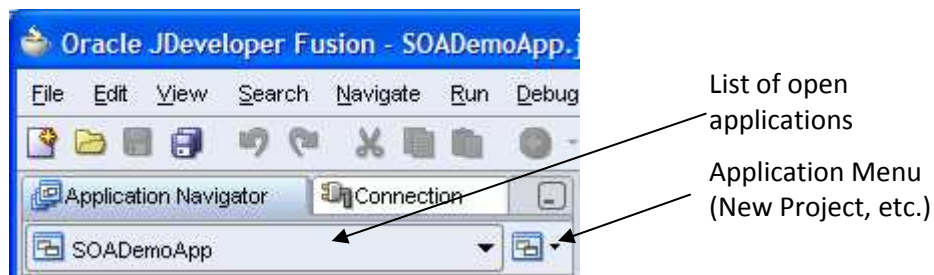
```
sqlplus soademo/soademo @create_creditrating_table.sql
```

7.5 Deploying CreditCardValidation composite

Open and deploy the Credit Card Validation composite.

1. In JDeveloper, select **View > Application Navigator**. If there are no currently open applications in JDeveloper you will see two options in the Application Navigator window: **New Application** and **Open Application**. Select **Open Application**.

Otherwise you will see an open application listed in the **Application Navigator** window and you will need to click on the open applications list and select **Open Application**.



2. In the dialog, browse to `c:\po\105-end2end-PO-Processing\Ch2\CreditCardValidation` and open `CreditCardValidation.jws`

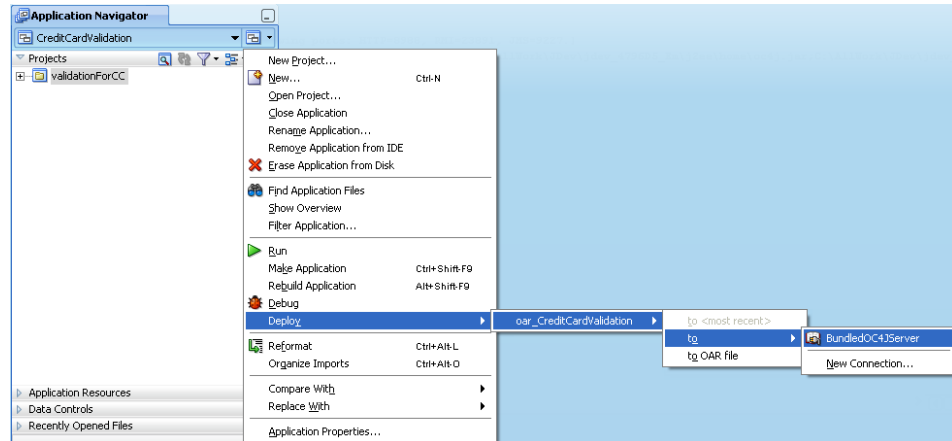
Now, start the server if it is not already started and deploy the Credit Card Validation application using the **Deploy** command.

3. Select the Run button in the toolbar to start the server. This button is enabled when an application is open in JDeveloper and you have selected Integrated OC4J from the Tools/Preferences/Run menu. If you are using a remote OC4J, just make sure that server is running.

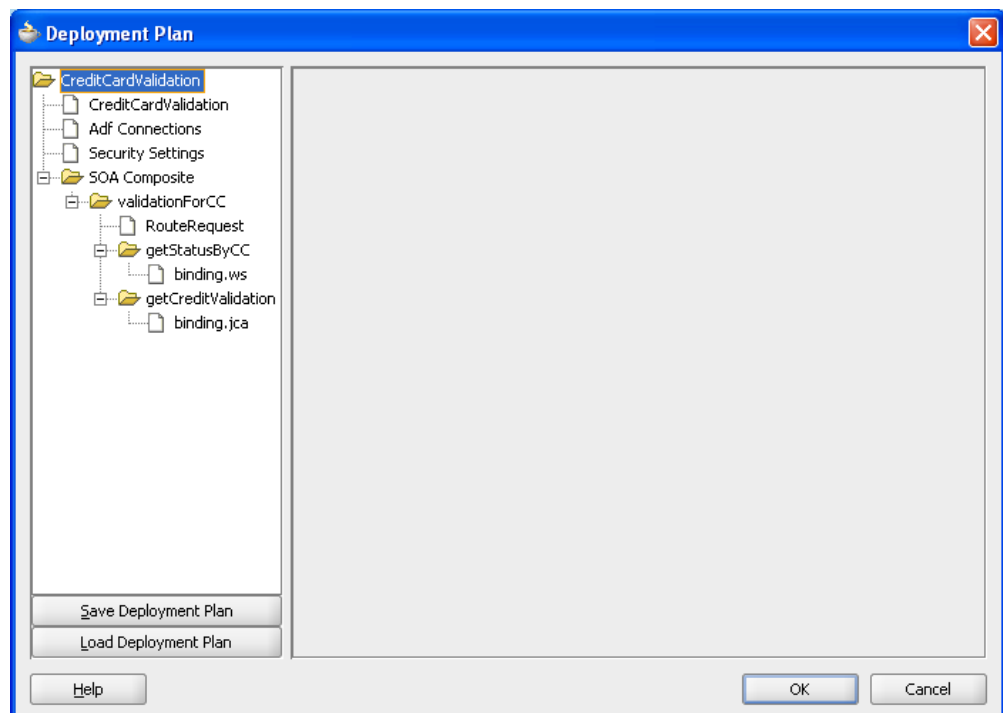


Wait for the server to finish starting before going on to the next step. The server is completely started when it posts the message, *Done deploying composites* in the log window.

4. Select the application name in the application navigator.
5. Then, in the application menu (see picture above to find and select the correct menu), select *Deploy...to...BundledOC4J*. Make sure you are using the application menu and not the project menu.



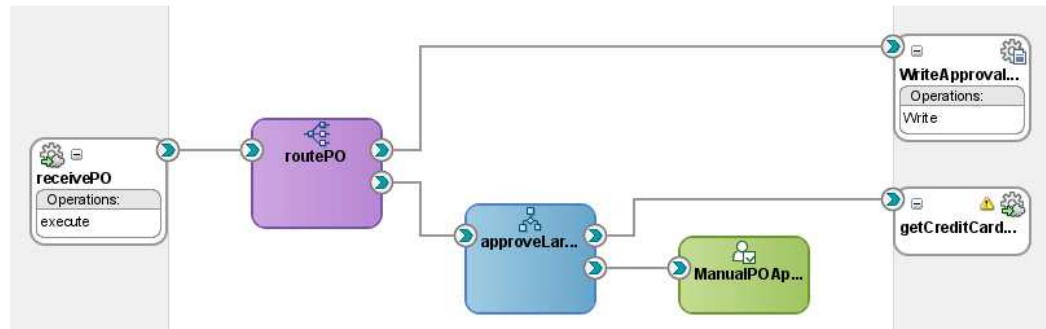
The deploy command will first compile the composite and then open the Deployment Plan.



6. Select *OK* and wait for the application to finish deploying. You will see messages in the server log when this is complete.

The next step is to open the PO Processing application but you do not need to deploy it. This application requires the path to the WSDL file of the Credit Card Validation application so the server must be running for it to find the WSDL.

7. Open the **POProcessing** Application just the same way you previously opened the Credit Card Validation application and expand the application to see the files within the application. Double-click on **composite.xml** to open that file.

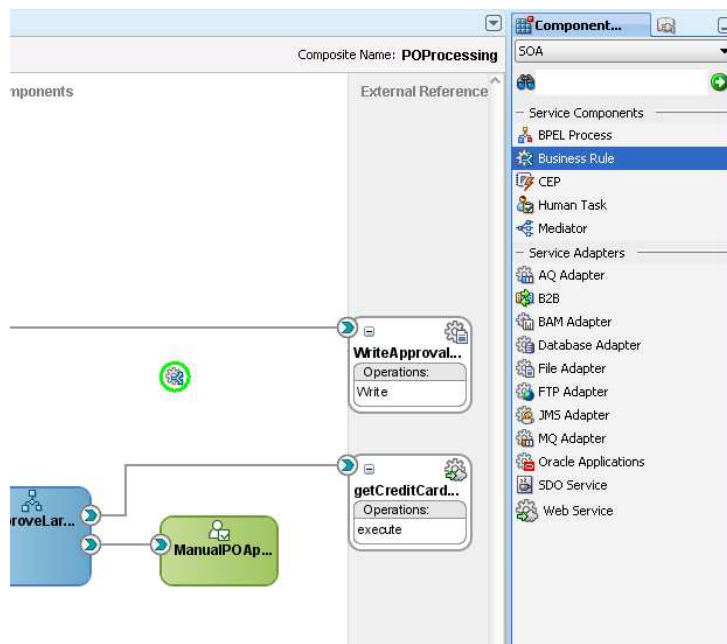


8. You may or may not see the warning icon on the **getCreditCard** web service reference. If you see this warning, make sure your server is currently started as in step 3 above. The warning icon means that JDeveloper can't find the WSDL file referenced in the web service reference. You do not need to resolve this now while you modify PO Processing to add the Business Rule but you will need to when you are ready to deploy the PO Processing application.

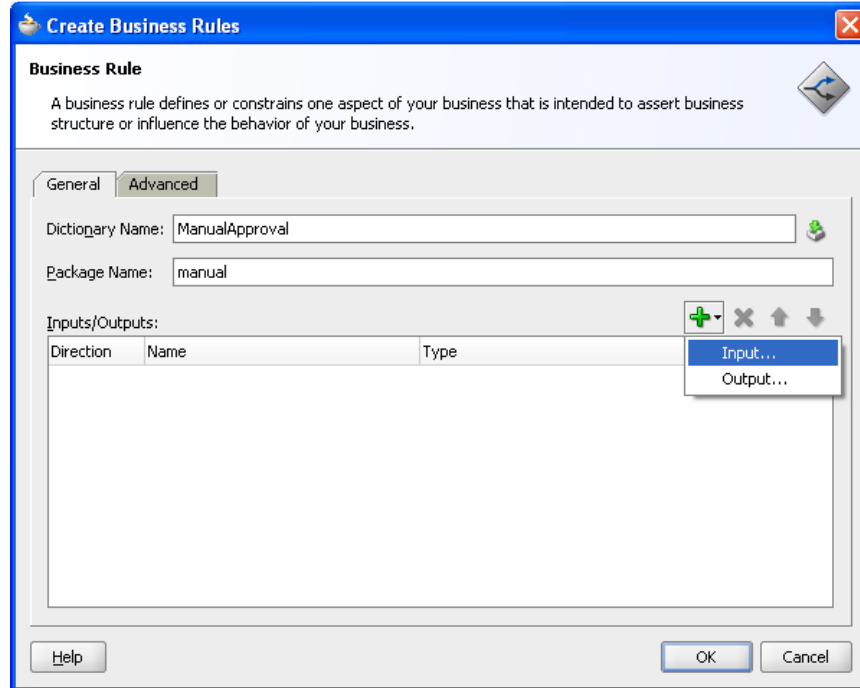
7.6 Adding a Business Rule to POProcessing

You are now ready to add the Business Rule. You first add the Business Rule component to the composite and then you invoke the rule from the BPEL process.

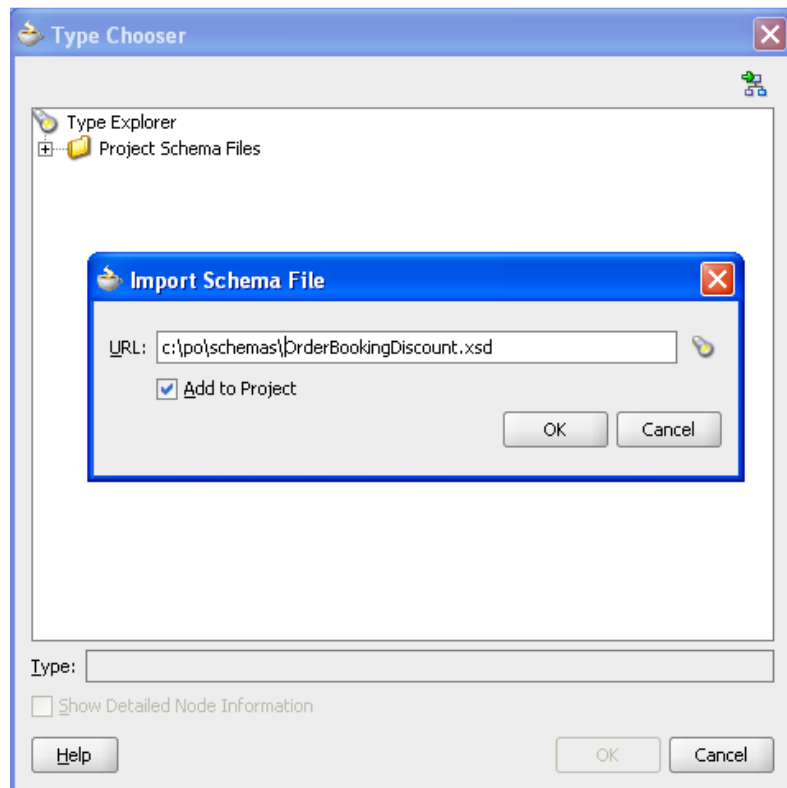
9. Add the Business Rule component to the composite by dragging it to the Components section.



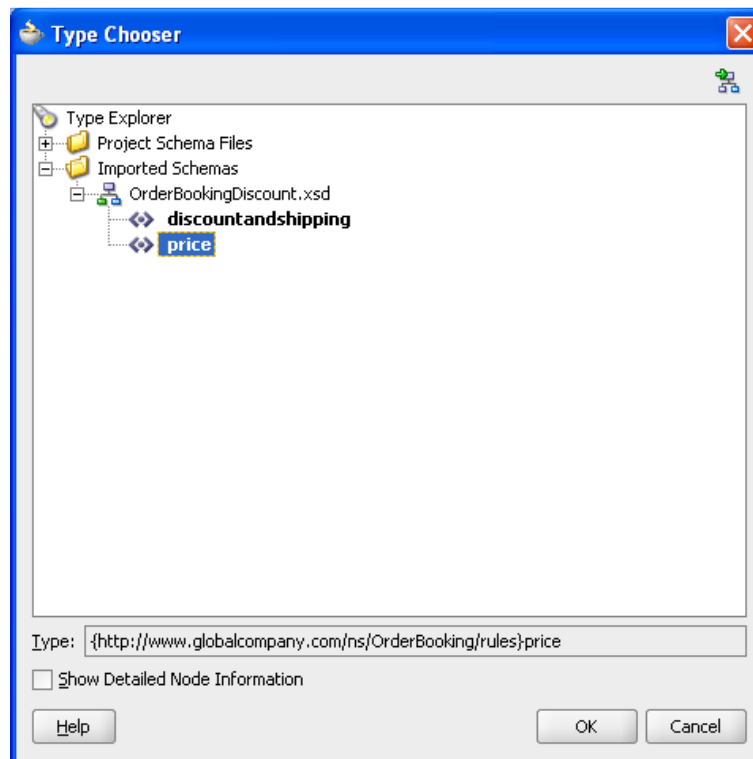
10. Complete the Create dialog by setting the Dictionary name and package name. Then set the input and output schema.



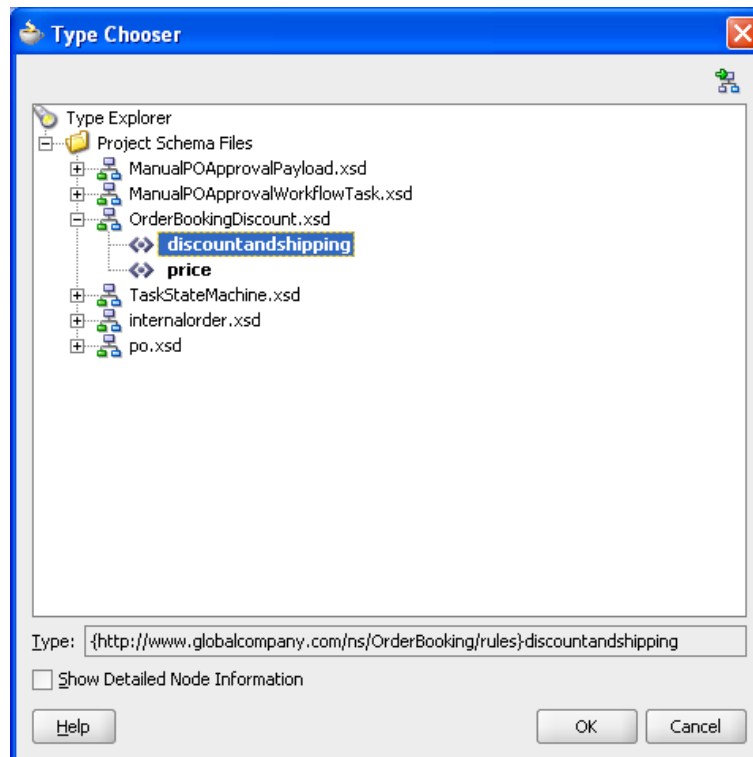
11. Selecting Input from the green plus-sign drop down opens the Type chooser for choosing the schema. Select the top right button to import a schema and then browse to the schema location: `c:\po\schemas\OrderBookingDiscount.xsd`



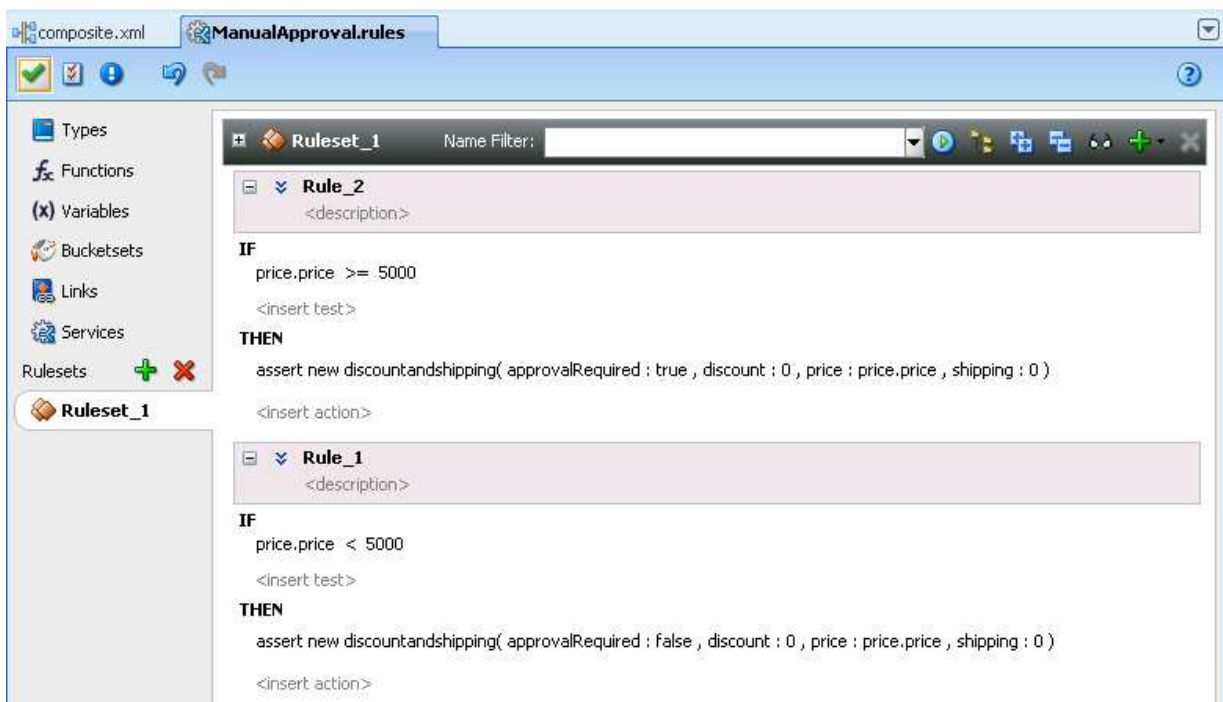
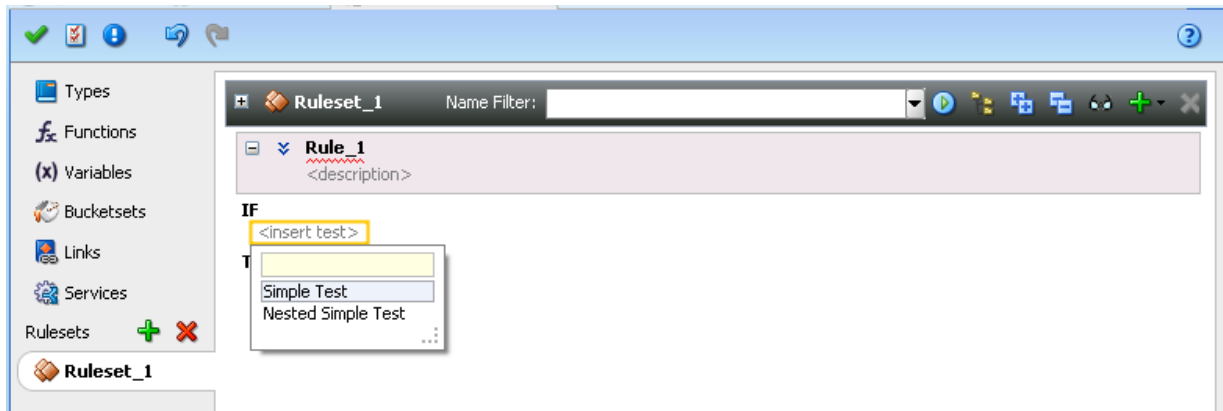
12. Select `price` for the input schema type.



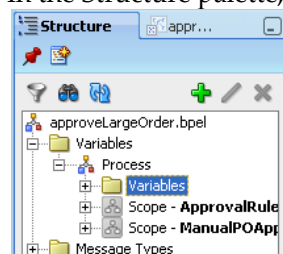
13. Repeat the steps for the output schema. This time you do not need to import the schema type. Just select the type `discountandshipping`.



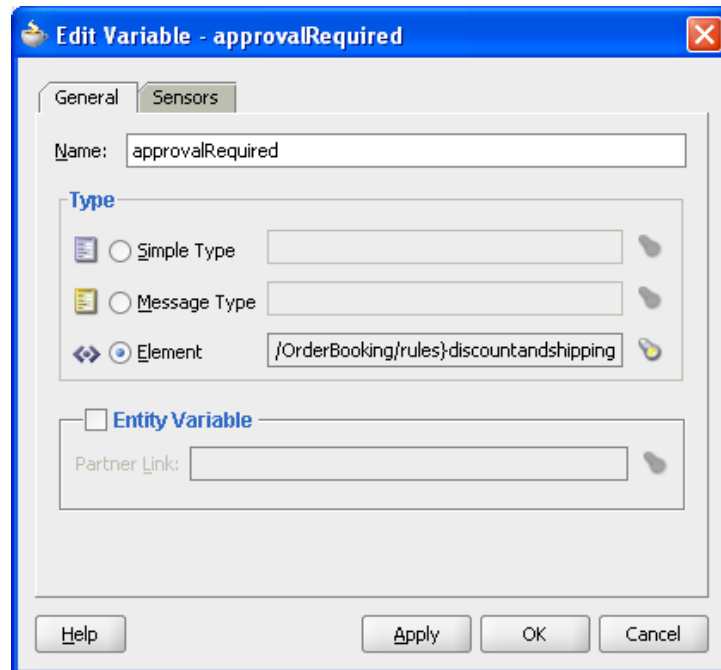
14. Once the dialog is complete, select OK to close it. Then double-click the Rules icon and create the RuleSet. Use the **Simple Test** selection and then complete the tests for ≥ 5000 and for < 5000 as shown below.



15. Save and close the Rules dialog. Next, double-click the BPEL icon to open the BPEL process. You will be adding a Rule activity to just before the existing Human Task. You first must create a variable to store the output from the RuleSet.
16. In the Structure palette, open Variables/Process and select the Variables node.

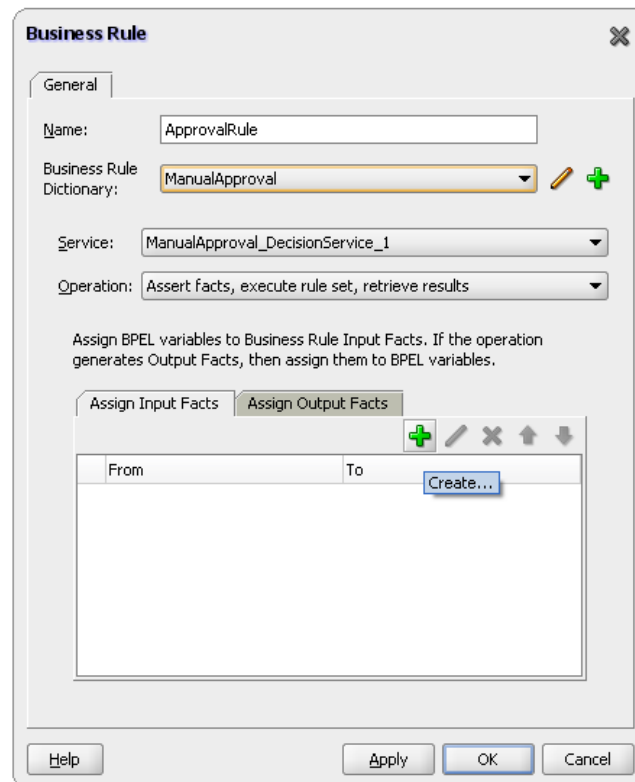


17. Select the green plus-sign to add a variable named `approvalRequired` of type `discountandshipping`.



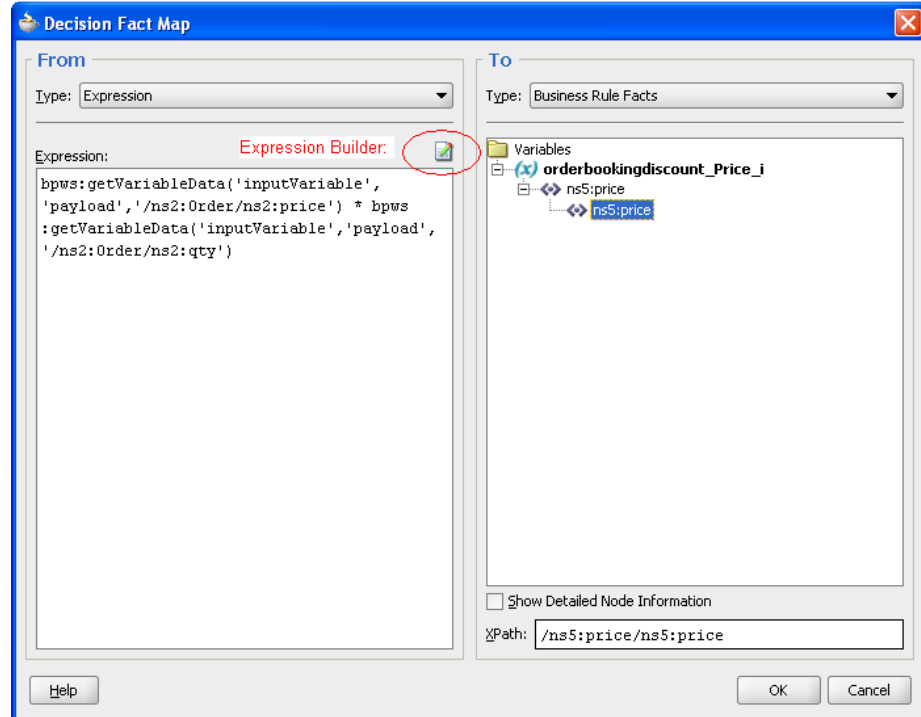
The dialog box titled "Edit Variable - approvalRequired" has two tabs: "General" and "Sensors". The "General" tab is active. It contains a "Name" field with the text "approvalRequired". Below it is a "Type" section with three radio buttons: "Simple Type", "Message Type", and "Element". The "Element" radio button is selected. To the right of the "Element" radio button is a text field containing the XPath expression "/OrderBooking/rules}discountandshipping". Below the "Type" section is an "Entity Variable" section with a checkbox that is not checked and a "Partner Link" field. At the bottom of the dialog are buttons for "Help", "Apply", "OK", and "Cancel".

18. Now drag the Business Rule to the location just before the Human Task and the Rule dialog will open. Set the Name and select the Business Rule Dictionary you just created in the composite. Next complete the Input Facts and Output Facts.

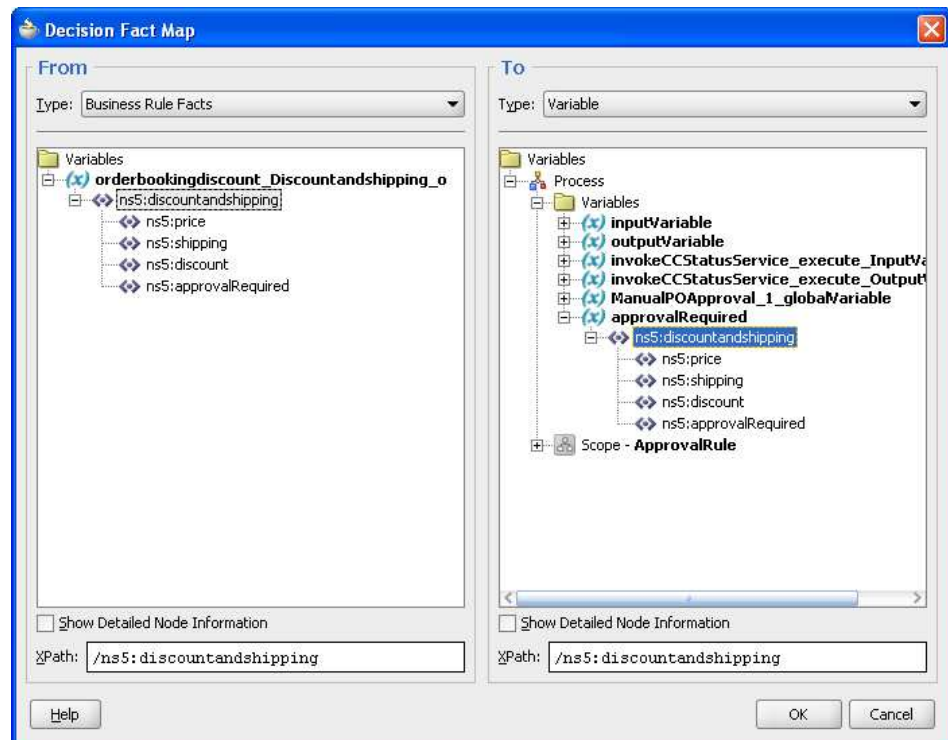


The dialog box titled "Business Rule" has a "General" tab. It contains a "Name" field with the text "ApprovalRule". Below it is a "Business Rule Dictionary" dropdown menu with "ManualApproval" selected. To the right of the dropdown are a pencil icon and a green plus icon. Below the dropdown is a "Service" dropdown menu with "ManualApproval_DecisionService_1" selected. Below the service dropdown is an "Operation" dropdown menu with "Assert facts, execute rule set, retrieve results" selected. Below the operation dropdown is a text area with the instruction: "Assign BPEL variables to Business Rule Input Facts. If the operation generates Output Facts, then assign them to BPEL variables." Below the text area are two tabs: "Assign Input Facts" and "Assign Output Facts". The "Assign Output Facts" tab is active. It contains a table with two columns: "From" and "To". Below the table is a "Create..." button. At the bottom of the dialog are buttons for "Help", "Apply", "OK", and "Cancel".

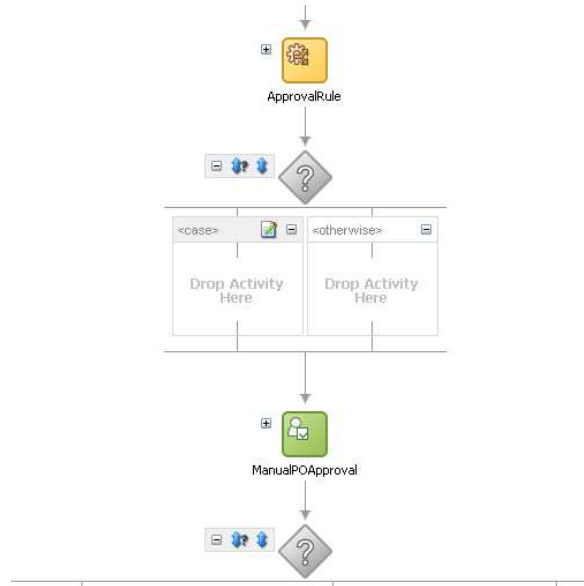
19. The input fact will be the price value. You create an expression using the expression builder to multiply the item price by the quantity.



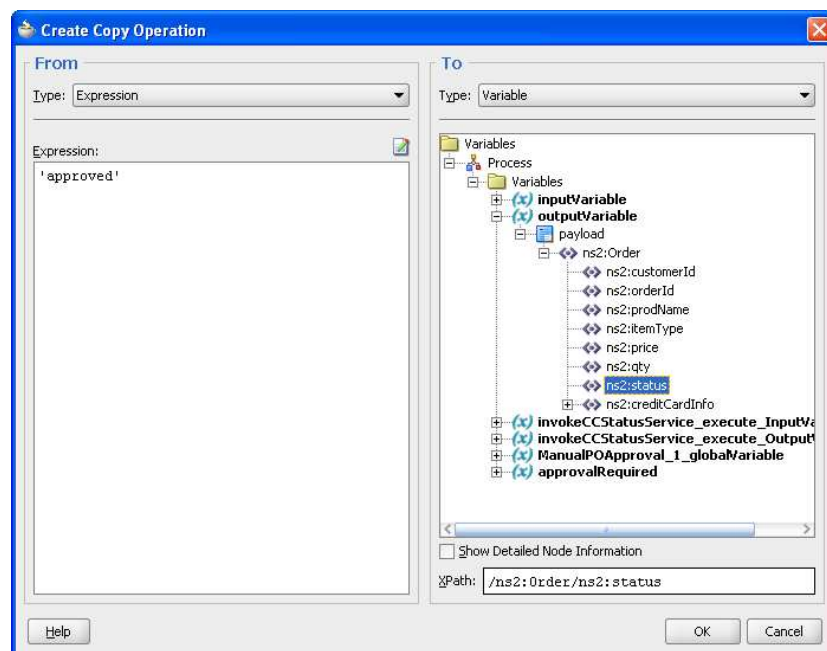
20. After the RuleSet is evaluated, the output fact is returned. Set the value to the variable you just created. Make sure to select the whole payload to copy all the values over (although we are only interested in the approvalRequired value).



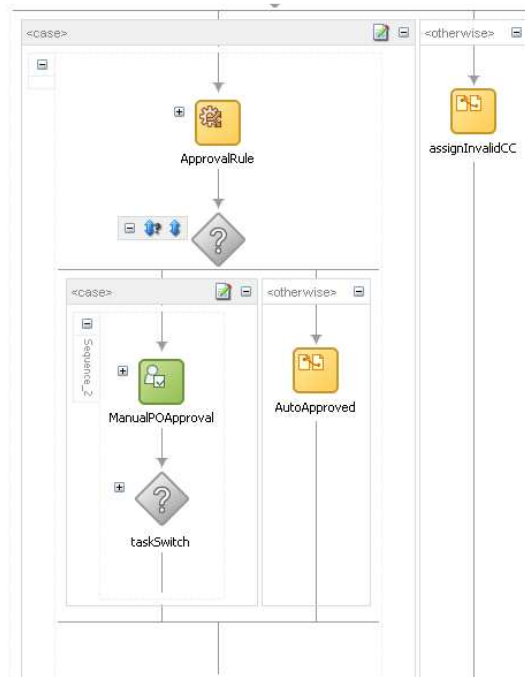
21. Now, copy a **Switch** activity to below the **Rule** activity and expand the case blocks.



22. Move the **Human Task** into the case block for this switch so that it only executes when the test case is true. There is a **Switch** activity that follows **Human Task** for processing the **Human Task** results which sets the status to **approved** or **rejected** accordingly. Move this into your new switch activity case statement as well.
23. Use the Expression Builder to set the test case expression in the case block:
`bpws:getVariableData('approvalRequired','/ns5:discountandshipping/ns5:approvalRequired') = 'true'`
24. Next create an assign activity in the **Otherwise** block of your new **Switch**. This assigns the status value **approved** for the case where the manual task is not required.



25. Your completed BPEL changes look like this.

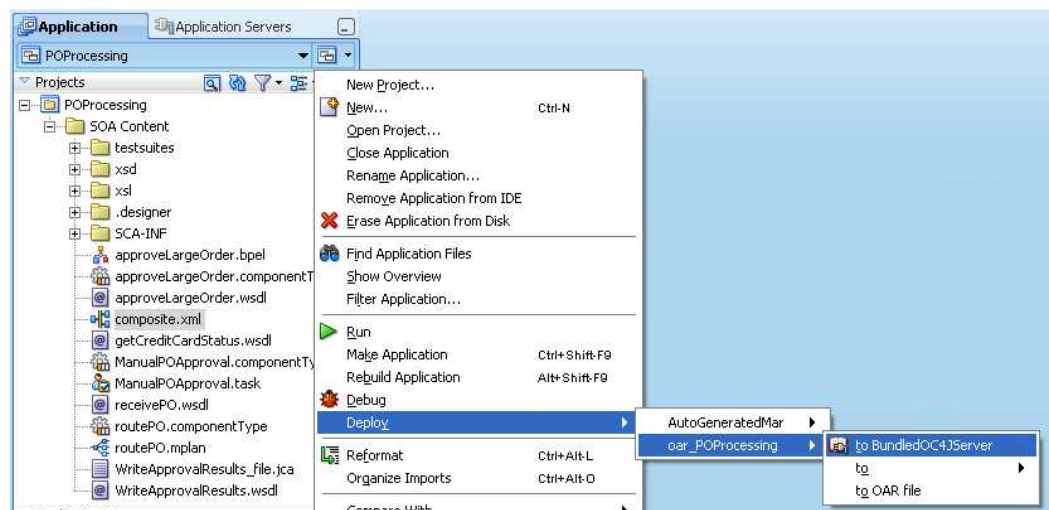


CAUTION: there is a bug to avoid here. Save and close your BPEL file. Open it again and you will see an identical Switch activity appear in the location from where you just moved it. It is a duplicate of the Switch that follows the Human Task and it must be deleted. Delete the duplicate (the one outside of your new Switch) and close and open the BPEL process again to confirm all is well.

You have completed all changes and are ready to deploy. Continue with the next section.

7.7 Deploying POProcessing composite

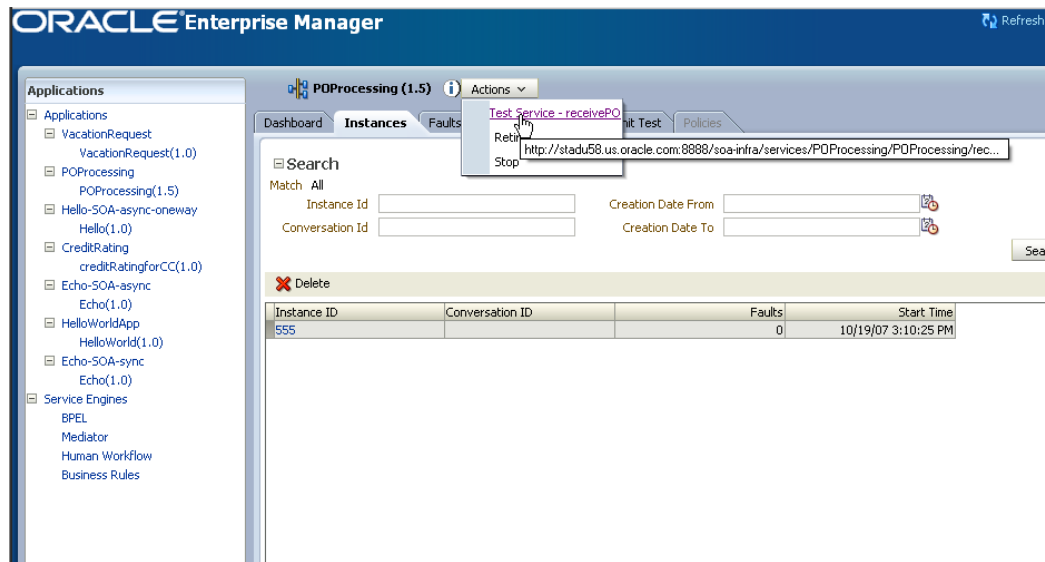
26. In the same way as you did with Credit Card Validation in step 5, deploy your newly updated POProcessing. Using the application menu, select [Deploy...BundledOC4J](#).



27. When you see the Deployment Plan dialog, select OK and wait for the application to finish deploying.

7.8 Running the application

28. Once both applications are deployed, you are ready to try running them. Open the SOA Console at <http://localhost:8988/SOAConsole> or <http://localhost:8888/SOAConsole> depending on your installation.
29. Click on **POProcessing** and in the **Actions** list choose to test your service.



30. Enter a small order by doing one of the following:

- Type the values into in to the HTML form, or
- Click **XML Source** so you can paste in the XML payload. This is the recommended way. Open the following file in a text editor:

`c:\po\input\po-small-Headsetx1.xml`

Copy the entire contents and paste them into the large text field in your browser:

receivePO endpoint

For a formal definition, please review the [Service Description](#).

Download the JavaScript Stub (BETA) for [execute_pt](#) and see its [documentation](#).

execute_pt

Operation: ☐ HTML Form ☒ XML Source

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://xmlns.oracle.com/ns/order">
    <ns1:PurchaseOrder>
      <ns1:CustID>1111</ns1:CustID>
      <ns1:ID>2222</ns1:ID>
      <ns1:productName>Bluetooth Headset</ns1:productName>
      <ns1:itemType>electronics</ns1:itemType>
      <ns1:price>49.99</ns1:price>
      <ns1:quantity>1</ns1:quantity>
      <ns1:status></ns1:status>
      <ns1:ccType>MC</ns1:ccType>
      <ns1:ccNumber>8765-8765-8765-8765</ns1:ccNumber>
    </ns1:PurchaseOrder>
  </soap:Body>
</soap:Envelope>
```

31. Click **Invoke**.

32. The **Test Result** screen won't have any response because this is a one-way invocation with no reply or callback. However the call to the File Adapter service will have resulted in a new file in `c:\temp`, called `order_n.txt`, where `n` is a sequence number like 1, 2, 3, etc.

You can open that file with a text editor and examine it (as shown in Figure 2).

Notice how field names have been translated by the mapping and are different from the input.

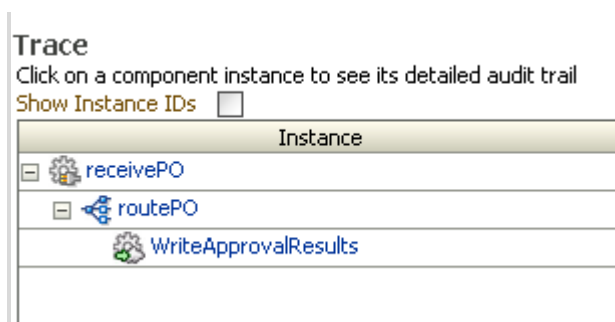
Figure 2 Output from order_n.txt file

```

1 <?xml version="1.0" ?><inpl:Order xmlns:inpl="http://xmlns.oracle.com/ns/order">
2   <inpl:customerId>1111</inpl:customerId>
3   <inpl:orderId>2222</inpl:orderId>
4   <inpl:prodName>Bluetooth Headset</inpl:prodName>
5   <inpl:itemType>electronics</inpl:itemType>
6   <inpl:price>49.99</inpl:price>
7   <inpl:qty>1</inpl:qty>
8   <inpl:status>approved</inpl:status>
9   <inpl:creditCardInfo>
10     <inpl:cardNumber>8765-8765-8765-8765</inpl:cardNumber>
11     <inpl:cardType>MC</inpl:cardType>
12   </inpl:creditCardInfo>
13 </inpl:Order>

```

33. You can also return to the SOA Console, click on the application or click the **Refresh** link. In the **Last 5 Instances** section click the most recent instance id to see the execution flow.



7.8.1 Additional Test Cases

There are four test cases for your application depending on the input data value of the total price:

- under \$1000 (price x quantity) – auto approval without BPEL component
- invalid credit card – auto reject using BPEL but no Rules or Task components
- 1000 to under 5000 - auto approval using BPEL and Rules but no Task
- 5000 and over – manual approval using BPEL, Rules but and Task components

The first case has been provided for you above. An invalid credit card number is:

4321-4321-4321-4321. When you get to the manual approval case, follow these instructions to see the human task.

34. Open the SOA Console to test the application using a very large input value in the same way as you did in the previously for a small order. Create a new order that's over \$5,000.
35. As usual, the **Test Result** page won't show any this is a one-way invocation with no reply or callback. Also notice, however, that a new order file was not created in `c:\temp`. That's because the application hasn't completed running yet. To see what's going on, you need to look at the instance audit trail.

Back in the SOA Console, click the latest instance (you may need to press the **Refresh** link in the upper right to update the list of instances.)

2. Click on the **approveLargeOrder** BPEL component.
3. Click the **Flow-Debug** tab to see the visual view of the BPEL process.
4. Scroll to the bottom and notice the BPEL process is incomplete, blocked on the Human Task until it is approved.
5. Now it is time to "change hats" and log on to the worklist application as a customer service representative to approve the task. Using a browser, navigate to <http://localhost:8988/integration/worklistapp> or <http://localhost:8888/integration/worklistapp> depending on your installation.
6. Logon using the following credentials:
 - **Username:** fmwadmin
 - **Password:** welcome1 or welcome (depends on the version of your installation)
7. The worklist application is an out-of-the-box application where users can view and manage their tasks.

Click on the most recent task to highlight it.
8. From the **Actions** menu at the top-right, select **Approve**.
9. Press the **Perform Action** button to the right of the **Actions** menu. This will approve the task and notify the BPEL process it can continue.
10. Return to the BPEL process flow. If you still have it open you can just refresh the page.

Notice that it is now complete.
11. You will also have a new order file in `c:\temp`.

Continue with different input data values and try all four test cases.

That's it! You have now installed, modified, and run the PO Processing application.