

9.2 Designing the flow

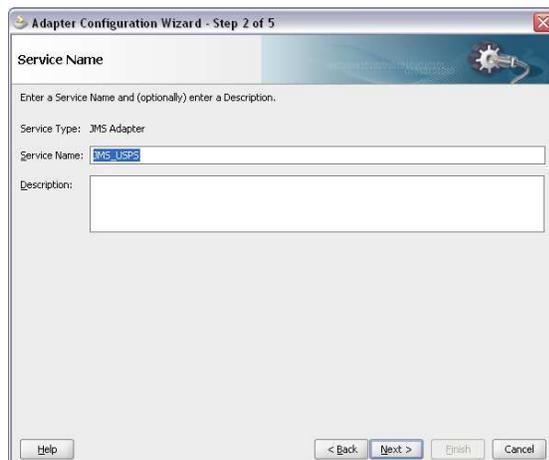
The fulfillment process was left in the last chapter after just having determined the carrier. To add the JMS messaging, add the three JMS services for USPS, UPS, and FedEx and then invoke the appropriate service from the BPEL process.

9.3 Add the JMS adapters

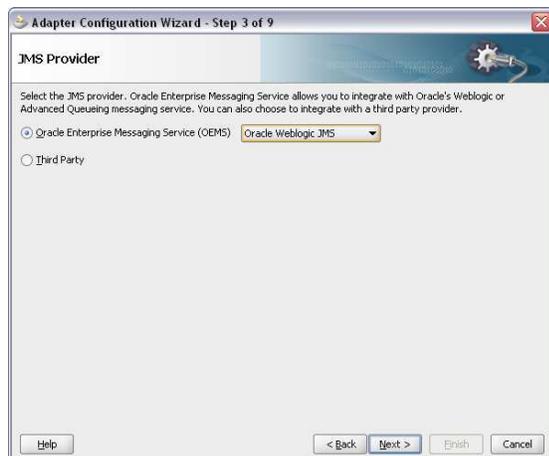
You first set up the JMS queue using the WebLogic console. You have to create a JMS destination queue, a JMS connection factory, and also a JMS connection pool. Follow instructions in Chapter 1 if you haven't already done so.

Next, you add a JMS adapter for each carrier. If you have your server running, the adapter wizard will be able to lookup the JMS destination queue.

1. In JDeveloper, open the POProcessing composite.xml.
2. Drag a **JMS Adapter** to the **External References** section of the composite.
3. Name the service: *JMS_USPS*



4. Click **Next**
5. Select the OEMS service *Oracle Weblogic JMS*



6. Click **Next**

7. Select the Service Connection

The screenshot shows the 'Adapter Configuration Wizard - Step 4 of 6' dialog box. The title bar includes a close button (X). The main heading is 'Service Connection'. Below the heading, there is a descriptive text: 'An Application Server Connection is required to configure this adapter. Select an application server connection already defined in your project or create a new connection.' Underneath, there are two radio buttons: 'Application Resources' (unselected) and 'Resource Palette' (selected). A 'Connection:' dropdown menu is set to 'MyAppServerConnection', with 'New...' and 'Edit...' buttons to its right. Below this, there are two text fields: 'User Name:' with the value 'weblogic' and 'Connect To:' with the value 'localhost:7001'. At the bottom of the dialog, there are four buttons: 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

8. Click Next

9. For the Adapter Interface, select the radio button for defining this later.

The screenshot shows the 'Adapter Configuration Wizard - Step 5 of 6' dialog box. The title bar includes a close button (X). The main heading is 'Adapter Interface'. Below the heading, there is a descriptive text: 'The adapter interface is defined by a wsdl that is generated using the operation name and schema(s) specified later in this wizard. Optionally, the adapter interface may be defined by importing an existing WSDL.' Underneath, there are two radio buttons: 'Define from operation and schema (specified later)' (selected) and 'Import an existing WSDL' (unselected). Below these, there are three input fields: 'WSDL URL:' (with a trash icon to its right), 'Port Type:' (a dropdown menu), and 'Operation:' (a dropdown menu). At the bottom of the dialog, there are four buttons: 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

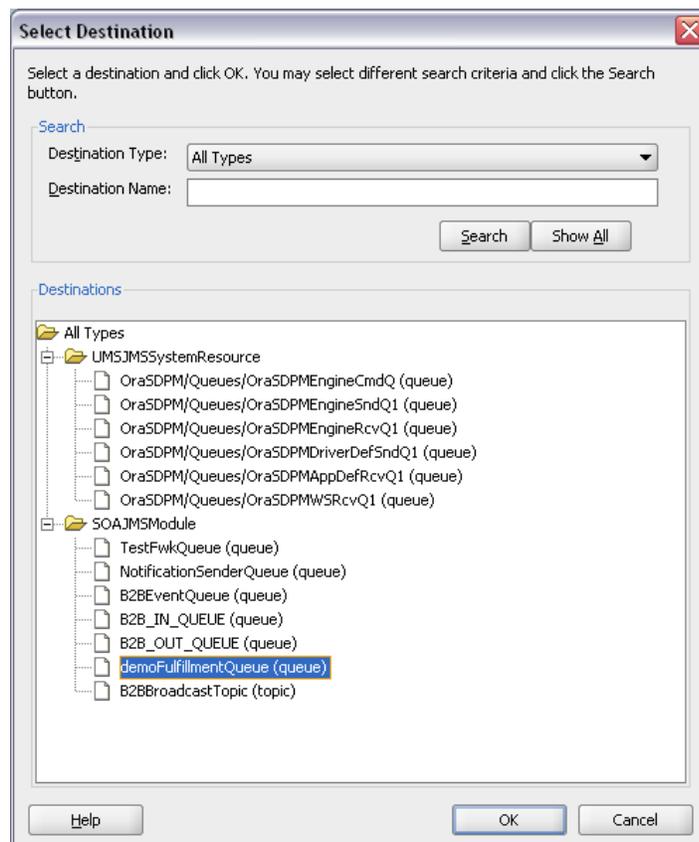
10. Click Next.

11. For the Operation, select Produce Message



12. Click Next

13. For the Produce Operation Parameters, click Browse to lookup the JMS destination queue and select *demoFulfillmentQueue*.



14. Click OK.

15. Enter the JNDI name associated with the JMS Adapter: *eis/Queue/demo*

Adapter Configuration Wizard - Step 7 of 9

Produce Operation Parameters

Enter the parameters for the Produce Message operation.

Destination Name (Queue):

Message Body Type:

Delivery Mode:

Priority:

TimeToLive:

Specify the JNDI name for the JMS Connection. The deployment descriptor for the deployed instance of the JMS Adapter must associate this JNDI name with a set of configuration properties needed by the JMS Adapter to access the JMS destination at runtime.

JNDI Name:

16. Click **Next**.
17. For the message schema, use the magnifying glass to browse to *fulfillment.xsd* and select the *Fulfillment* element.

Adapter Configuration Wizard - Step 8 of 9

Messages

Specify the schema that defines the message payload of the JMS destination. Specify the Schema File location and select the Schema Element that defines the message. Use the Browse button to find an existing schema definition. If you check 'Schema is Opaque', then you do not need to specify a Schema.

Native format translation is not required (Schema is Opaque)

URL:

Schema Element:

18. Click **Next** and then click **Finish**.

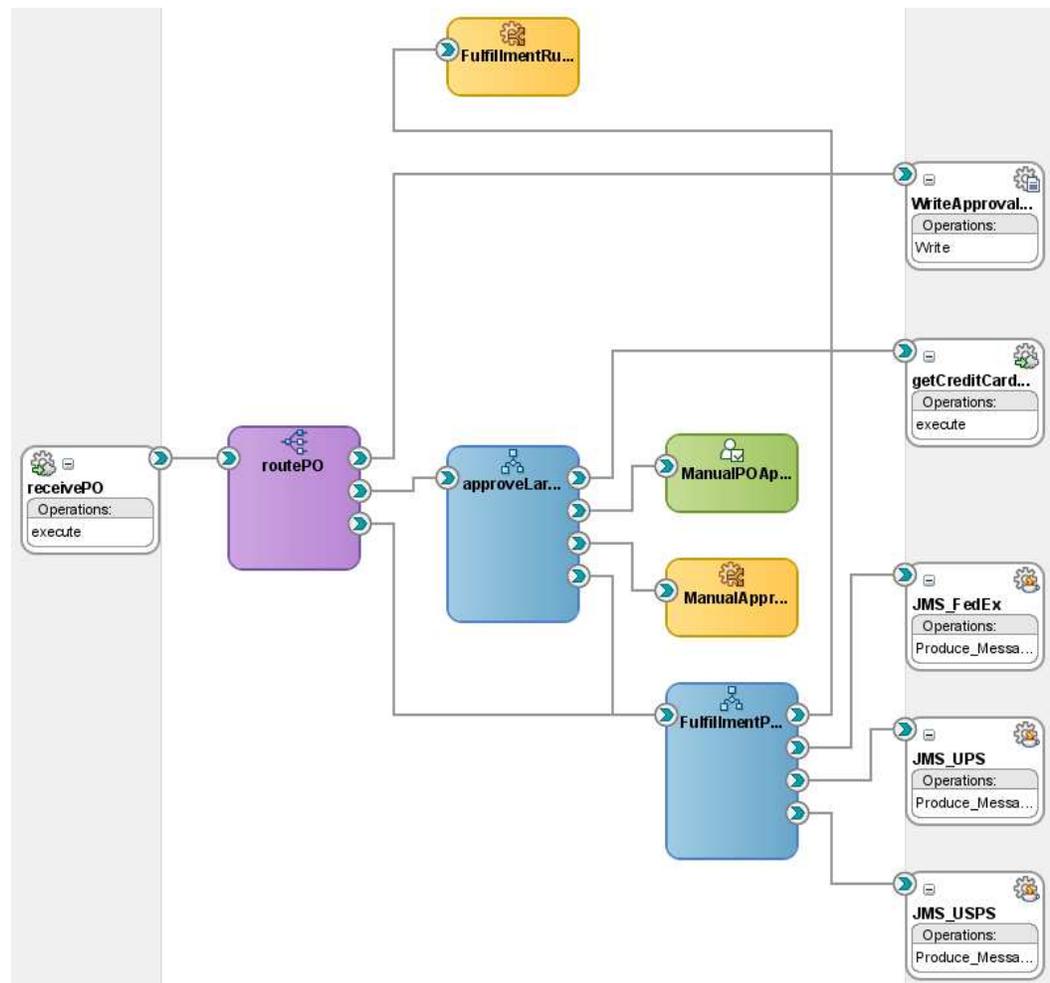
The first adapter service is complete.

19. Repeat the process to create two more services, *JMS_UPS* and *JMS_FedEx*.
20. From *FulfillmentProcess*, drag a wire to each of the three JMS services.

These services represent the three carriers and the message that is sent is the order information for fulfillment.

As mentioned earlier, normally the three carriers would all have individual destination queues but for easier testing, just use one queue for all of them.

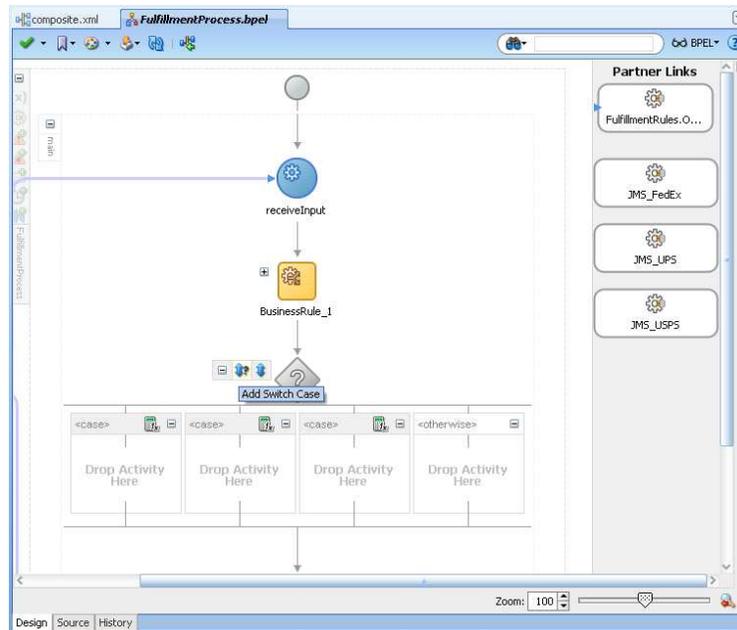
Your composite now looks like this.



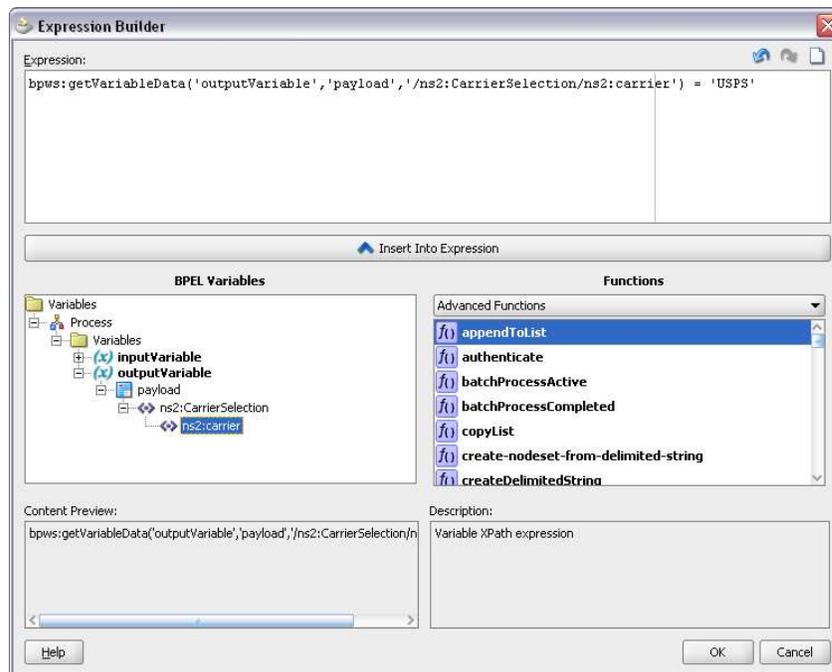
9.4 Invoke the services from BPEL

Next go into the BPEL process and invoke the three services according to the value of carrier returned from the decision service.

21. Open the *FulfillmentProcess* BPEL process and drag a **Switch** activity below the *FulfillmentRule* **Business Rule** activity.
22. Expand the **Switch** and add two more switch cases.

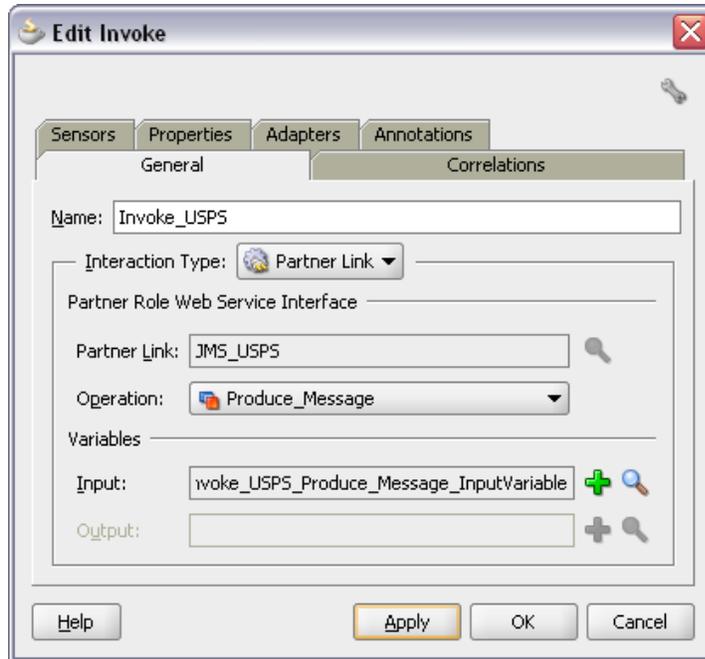


23. Double-click on the <case> bar and set the **Name** to *USPS*.
24. Select the **Expression Builder** icon to set the condition expression for the first case. Enter the expression as follows
`bpws:getVariableData('outputVariable','payload','/ns2:CarrierSelection/ns2:carrier') = 'USPS'`

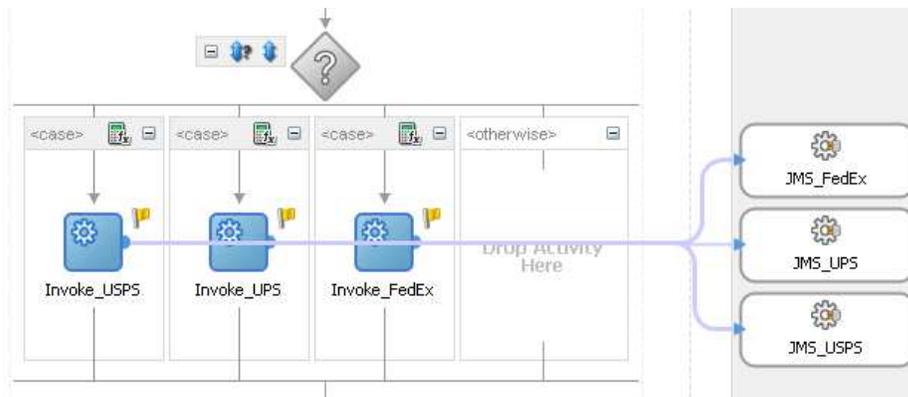


25. Click **OK**. Repeat for the next two cases, setting the condition for *UPS* and *FedEx* respectively.

26. Drag an **Invoke** activity into the first case for USPS. Drag a wire from the **Invoke** to the *JMS_USPS* partner link. The **Invoke** dialog opens.
27. Set the name to *Invoke_USPS* and create the input variable using the default values. Click **OK**.



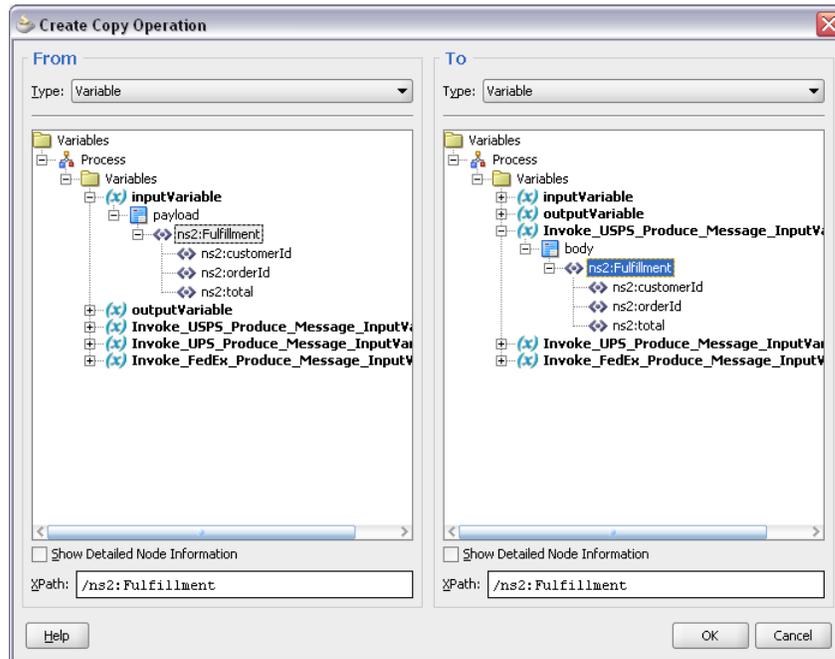
28. Repeat for UPS and FedEx.



Now you need to assign data to the input variables for the three service invokes.

29. Drag an **Assign** activity into the first case just before the **Invoke**.

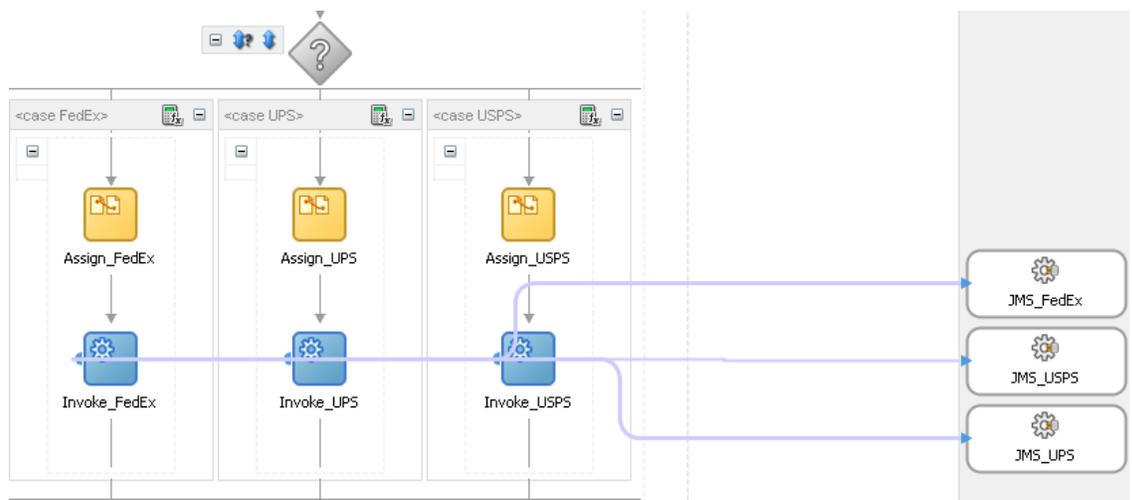
30. Open the **Assign** and create a copy operation to copy the fulfillment data to the input variable for the USPS service.



31. Repeat for the remaining two services.

The final case, the **otherwise** case, can be handled in one of two ways. You could choose to do something at this point such as raise a fault. That would be appropriate if in the future there is a fourth carrier setting returned from the rule and you wanted to be sure to flag it so that code could be added to handle it. Instead, for now just do nothing. You can either remove the otherwise block or put an **Empty** activity there.

32. Remove the **otherwise** case block to do nothing in the otherwise case.



And that completes the JMS addition to the fulfillment process. Give it a try.

9.5 Deploying the application

Deploy the *POProcessing* in the same way as before using the **Deploy** command on the Project Menu. Read **Appendix A Deploying and Running a Composite Application** to refresh your memory on how to deploy if you need to.

9.6 Testing the application

Again, use the flow view in the EM console to see that everything is working correctly.

1. Open EM console and click *POProcessing*. Click **Test**.
2. Using the input files you will find in `c:\po\input` run three tests, with small, large, and extra large orders.
3. View the instance data using EM console. For the extra large order, you have to submit the approval task to complete the process.
4. Verify that the *FulfillmentRules* returns USPS, UPS, and FedEx as the delivery carrier appropriately.
5. Verify the JMS message is sent appropriately by viewing the instance flow details and the message using one of the following three methods.
 - (i) WLS Console
 1. Open **Services/Messaging/JMS Modules/SOAJMSModule**
 2. Click *demoFulfillmentQueue/Monitoring*
 3. Select checkbox for *SOAJMSModule!demoFulfillmentQueue*
 4. Click **Show Messages** to view the messages waiting in the queue.
 - (ii) ConsumeJMSFulfillmentApp
 1. Go to the `c:\po\solutions\ch9` directory and deploy the *ConsumeJMSApp* composite. This application has been set up to listen on the JMS queue used in this lab. Whenever a message shows up, this composite fires off an instance that you can view in the EM console using Flow Trace and Flow Details. It also writes a file.
 2. When you no longer want this application to consume messages (it polls as long as it is running), use EM to shut down the composite. The **Shut Down** button is next to the **Test** button. You can **Start Up** again later as desired.
 - (iii) OEMS Send/Receive utility
 1. Locate *jms-send-receive.zip* in the downloads directory for this course.
 2. Unzip and use the Receive utility to view the messages in the queue.

9.7 Operations and naming

This section gives you all of the operations and names for objects created in this chapter. Experienced users can use this for creating the objects in this chapter quickly. Any questions on details for a particular operation listed here can be found in the preceding sections. The information is divided by the sections in this document.

Add the JMS adapters

- In composite: **JMS Adapter:** JMS_USPS
- **OEMS service** Oracle Weblogic JMS
- **Service Connection:** MyApplicationServer
- **Define this Adapter Interface later**
- **Operation:** Produce Message,
- **Produce Operation Parameters:** demoFulfillmentQueue.
- **JNDI name:** eis/Queue/demo
- **Message schema:** fulfillment.xsd > Fulfillment
- Repeat for: JMS_UPS and JMS_FedEx, wire all from FulfillmentProcess

Invoke the services from BPEL

- *FulfillmentProcess* BPEL process > **Switch** below the **Business Rule**
- Expand the **Switch** and add two more switch cases.
- Double-click on the <case> bar and set **Name:** USPS
- Expression:
`bpws:getVariableData('outputVariable','payload','/ns2:CarrierSelection/ns2:carrier') = 'USPS'`
- Repeat for the next two cases, setting the condition for 'UPS' and 'FedEx' respectively.
- **Invoke:** Invoke_USPS inside case block, wire to the JMS_USPS
- **Input variable:** defaulted
- Repeat for UPS and FedEx.
- **Assign:** Assign_USPS just before **Invoke_USPS**.
- Copy From: inputvariable > fulfillment ,
- Copy to: Invoke_USPS_Produce_Message_InputVariable.
- Repeat for UPS and FedEx.

The application is completed. Continue with Section 9.5 above to deploy and test your application.