



SupportPac LA71: IBM Operational Decision Manager Integration for WebSphere Process Server

Getting started with IBM Business Process Manager

Task 7 – Integration Designer assembles an Advanced Integration Service containing an SCA Decision Component.

Copyright

Copyright notice

© Copyright IBM Corp. 1987, 2012

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Trademarks

IBM, the IBM logo, ibm.com, WebSphere, and ILOG are trademarks or registered trademarks of International Business Machines Corp., in many jurisdictions worldwide.

Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at *Copyright and trademark information*.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Overview

In this task you will use the principles that you have learnt in previous tasks to integrate a Managed Decision Component back into the original BPMN process. You will then go through the steps a Business Process Author undertakes to test and change the decisions in that process in task 8.

Readers who want to skip the exercise should import the task 7 solution into Process Center from the snapshot saved at

[SupportPac LA71 Path]\BPMTutorial\task7\Insurance_Sample - LA71TT7.twx

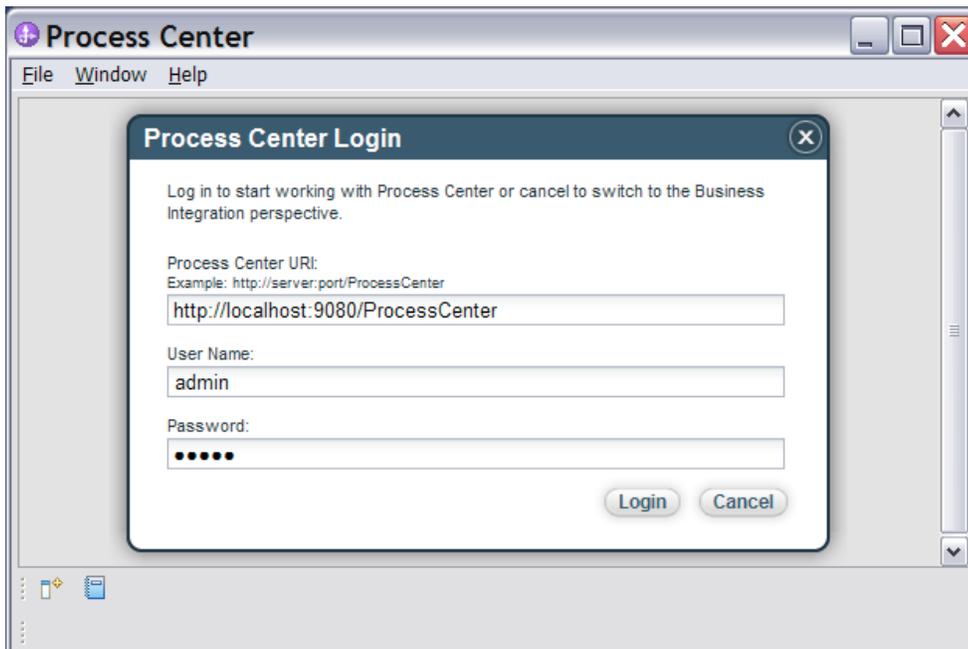
You should then open the Process App in the IID workspace and test the integration as described in step 5.

Step 1. Import the Advanced Integration Services into the IID workspace.

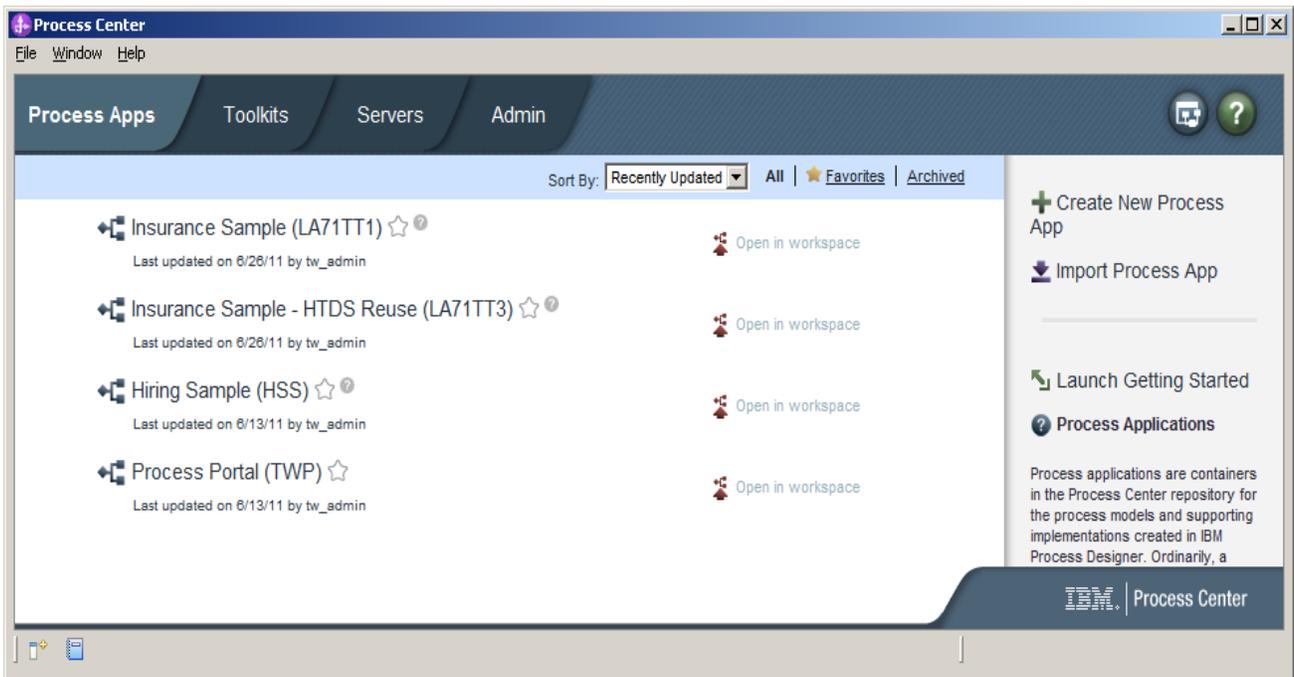
In Task 1 the Business Process Author designed the Decisions that were required and specified Advanced Integration Services interfaces that should be used to invoke the Managed Decision services that IT should develop. In this step you will bring the definition of those interfaces into your workspace.

From within your Integration workspace switch to the Process Center perspective.

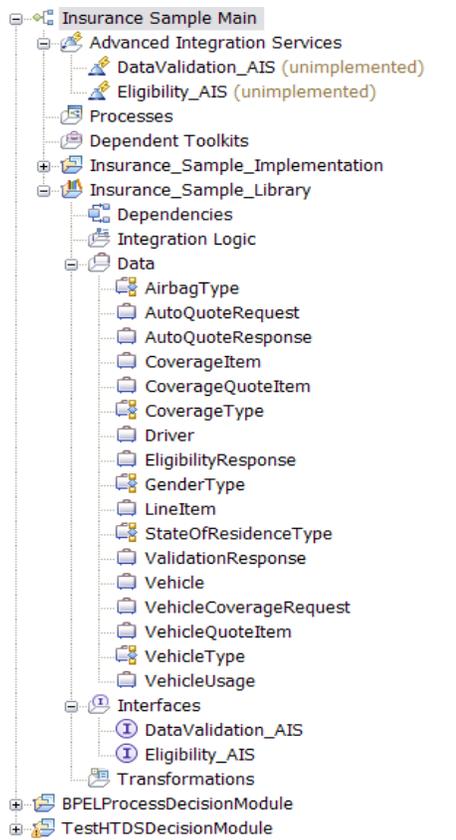
At the Process Center login enter the **Process Center URI** to correspond to your environment. Enter the User Name and Password – use admin / admin for a default install. Click **Login**.



Close the Welcome pane and select the Insurance Sample (LA71TT1) Process Application.



Select Open in workspace and the project will be opened showing the default structure of the two Advanced Integration Services.



If the Insurance_Sample_Implementation and Insurance_Sample_Library are not visible you would need to press the **Switch Process Applications and Toolkits To Detailed Mode** icon.



Note the two AIS interfaces (DataValidation_AIS and Eligibility_AIS) and the Business Objects that have been defined in the Insurance Sample Library. These Business Objects are the same as those used in the BRMS Decisions so we will not have any Data mapping to do.

Step 2. Create the AutoQuoteDecisionService Managed Decision.

In this step we will use the techniques described in task 5 to create an SCA Managed Decision Module from the Managed RuleApp generated in Task 2.

Select the **Insurance_Sample_Implementation** Module.
Right click and select **SupportPac LA71 > Create SCA Component from RuleApp**

The wizard opens with the **RuleApp Selection** screen.

Ensure the **Project** field is set to **Insurance_Sample_Implementation**

In the **Click Project or Archive to select RuleApp** field click **Archive...**

Browse to the directory where you saved the RuleApp archive or the completed **[SupportPac LA71 Path]\BPMTutorial\task2** directory and select **InsuranceSampleRuleApp.jar**. Ensure that the checkboxes next to the **DataValidation** and **Eligibility** rulesets are checked.

Business Decision Wizard

RuleApp Selection
Please select the ruleset(s) you would like to include.

Click Project to select a Module Project

Click Project or Archive to select RuleApp

Select the ruleset(s) to be included in the new business decision:

DataValidation
 Eligibility

Input Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationRespo...	ValidationResponse

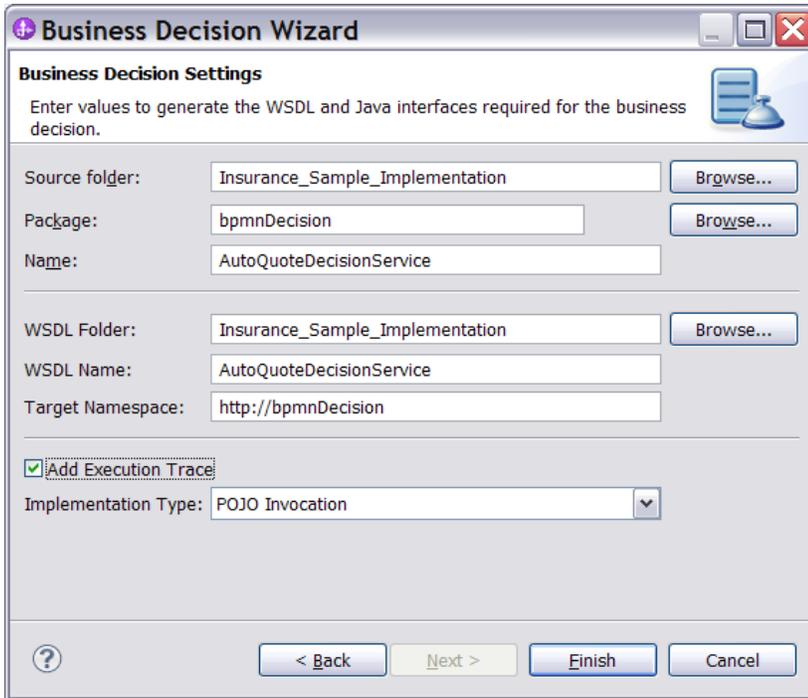
Output Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationRespo...	ValidationResponse

Click **Next >**. The **Object Model Mapping** screen appears and as all Business Objects have already been imported through the AIS interfaces no further mapping is required.
Click **Next >**.

In the **Business Decision Settings** screen:
Set the **Package** to **bpmnDecision**.
Set the **Name** to **AutoQuoteDecisionService**

Check the **Add Execution Trace** box to add execution trace to the implementation.
In the **Implementation Type** select **POJO Invocation**.



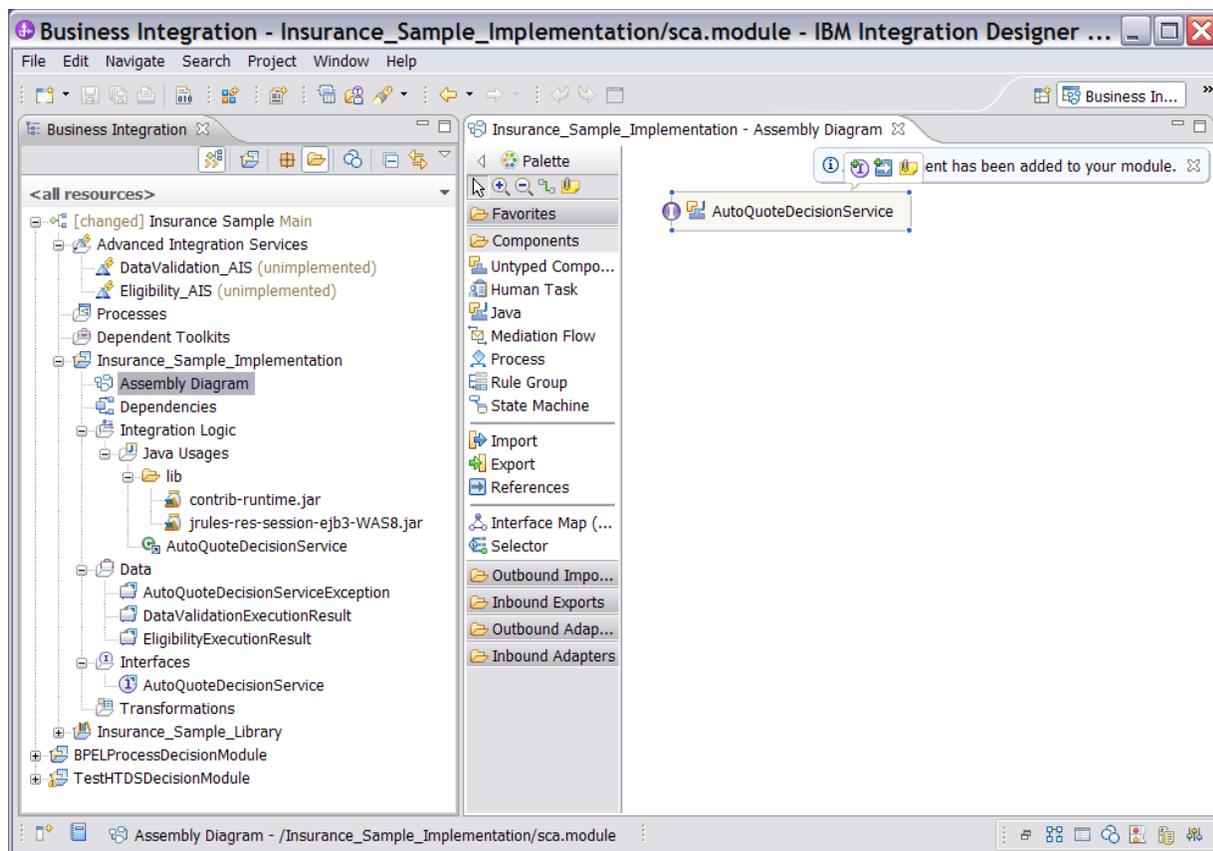
The screenshot shows the 'Business Decision Wizard' dialog box with the following settings:

- Business Decision Settings**: Enter values to generate the WSDL and Java interfaces required for the business decision.
- Source folder**: Insurance_Sample_Implementation
- Package**: bpmnDecision
- Name**: AutoQuoteDecisionService
- WSDL Folder**: Insurance_Sample_Implementation
- WSDL Name**: AutoQuoteDecisionService
- Target Namespace**: http://bpmnDecision
- Add Execution Trace**
- Implementation Type**: POJO Invocation

Navigation buttons at the bottom: < Back, Next >, Finish, Cancel.

Click **Finish**. The wizard completes and generates the **AutoQuoteDecisionService** SCA implementation.

Double click **Insurance_Sample_Implementation > Assembly Diagram** to check the structure of the implementation as shown below.



This service component is equivalent to that generated and tested in task 4 but is ready to use in the BPMN process specified by the Business Process Author.

Step 3. Implement Advanced Integration Services.

In this step you will implement the Advanced Integration services and bind them to the AutoQuoteDecisionService Managed Decision Component.

Select **Insurance Sample > Advanced Integration Services > DataValidation_AIS**

Right Click and select **Implement**.

In the **Select an Implementation Type** select **Microflow**.

Click **Finish**.

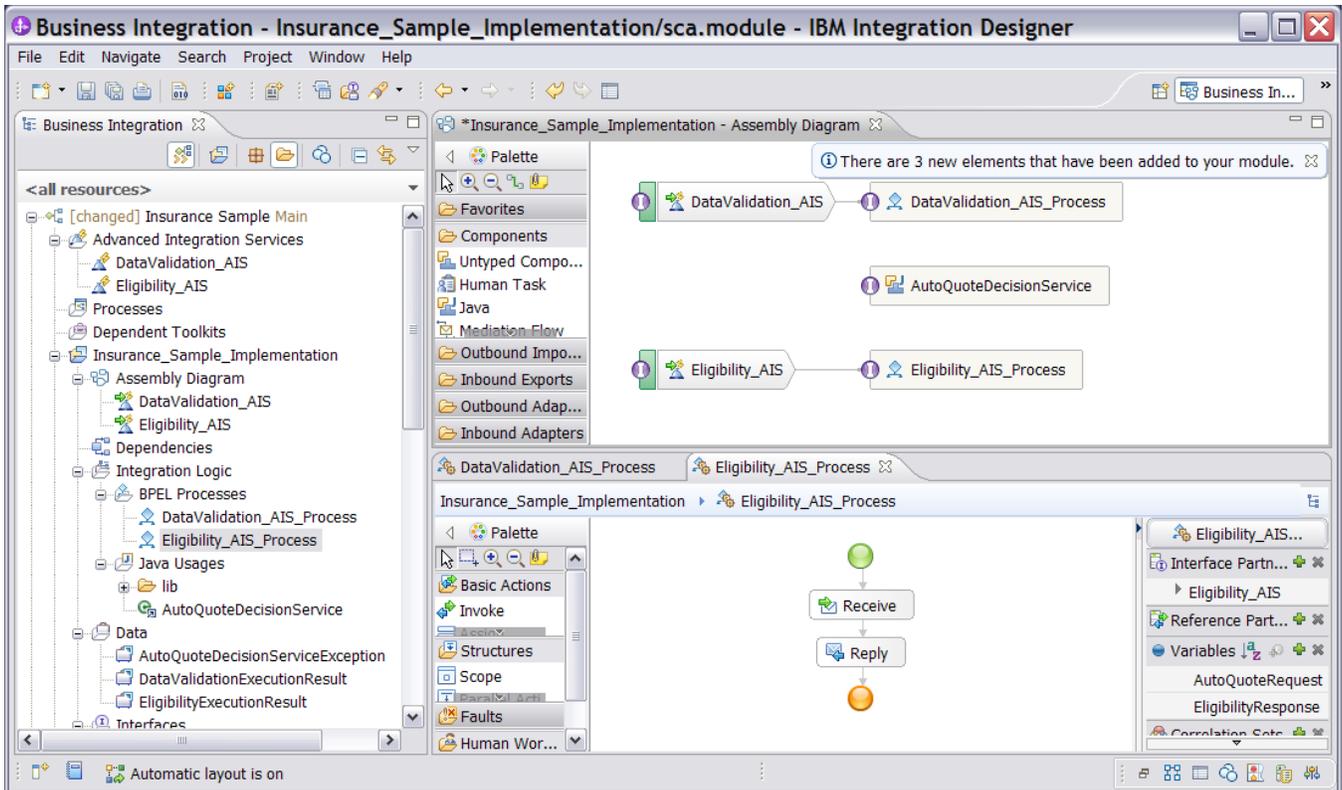
Select **Insurance Sample > Advanced Integration Services > Eligibility_AIS**

Right Click and select **Implement**.

In the **Select an Implementation Type** select **Microflow**.

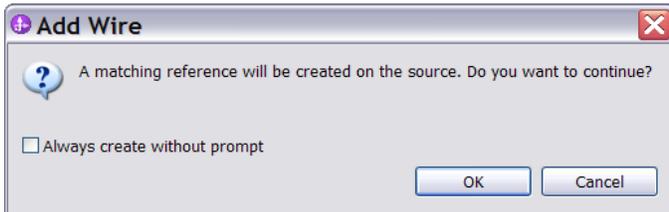
Click **Finish**.

This creates microflow BPEL processes for each AIS and integrates them into the assembly diagram as shown below. You may need to right click in the Assembly Diagram canvas and select Automatic Layout to ensure the elements are laid out nicely.



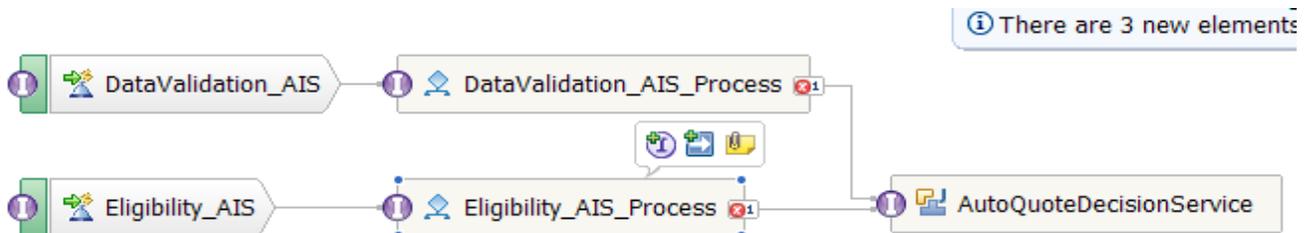
You now need to bind the microflows to the decision service.
 Select the **DataValidation_AIS_Process** and drag a wire from it to the **AutoQuoteDecisionService**.

Click **OK** to the popup.



Select the **Eligibility_AIS_Process** and drag a wire from it to the **AutoQuoteDecisionService**.
 Click **OK** to the popup.

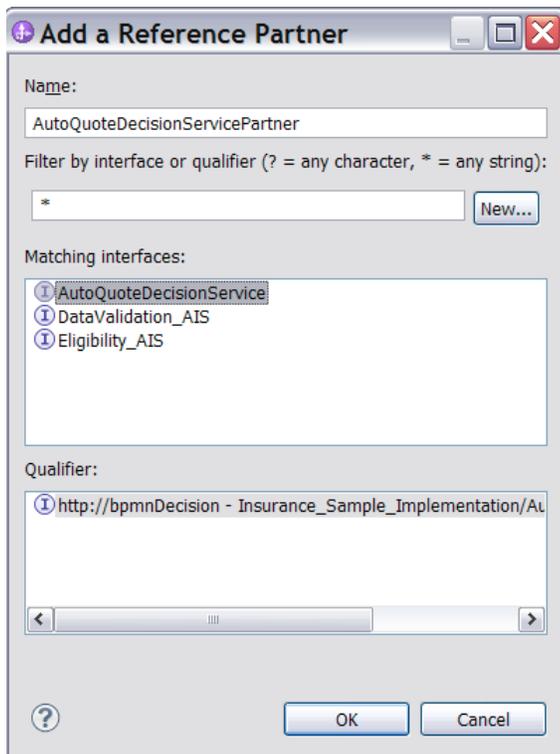
After saving and rebuilding the assembly diagram now looks like this. There are errors here as the microflows do not yet invoke the decision service.



Double-click **Insurance_Sample_Implementation > Integration Logic > BPEL Processes > DataValidation_AIS_Process** and click the '+' symbol on the Reference Partners palette to add a new Reference Partner.

For **Name** type **AutoQuoteDecisionServicePartner**.

For **Matching interfaces** select the **AutoQuoteDecisionService**.



Click **OK**.

In the Process Editor Select **Basic Actions > Invoke** and click between the **Receive** and **Reply** activities.

Select the Invoke Element and Properties screen in the Details tab.

Click the **Browse...** button for the Partner field, select **AutoQuoteDecisionPartner** in the dialog, and click **OK**.

Ensure the interface is **AutoQuoteDecisionService**.

For Operation select **DataValidation**.

Further down, click on **(none)** under Read from Variable next to AutoQuoteRequest and select **AutoQuoteRequest** in the popup, and click on **(none)** next to ValidationResponse and select **ValidationResponse** from the popup.

Under Store into Variable, click on **(none)** and select **New...** Click OK in the dialog that appears.

Task Flows Properties Build Activities Problems Server Logs

Invoke - Invoke

Description Partner:* AutoQuoteDecisionServicePartner

Details Interface:* [AutoQuoteDecisionService](#)

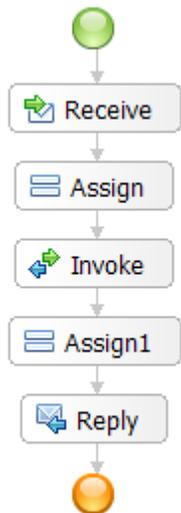
Server Operation:*

Use data type variables mapping

	Name	Type	Read from Variable
Inputs	AutoQuoteRequest	AutoQuoteRequest	AutoQuoteRequest
	DataValidationResponse	ValidationResponse	ValidationResponse
	Name	Type	Store into Variable
Outputs	DataValidationExecutionResult	DataValidationExecutionResult	DataValidationExecutionResult

Two last steps are needed to initialize the data for the request and map the response to the reply. In the process editor, select **Basic Actions > Assign** from the palette and click between the **Receive** and **Invoke** activities. Select **Basic Actions > Assign** from the palette and click between the **Invoke** and Reply activities.

The Process Editor looks like this now.



Select the **Assign** activity and go to the Details tab of the Properties view. Click **Select From**, select **String (enter a value)** from the popup, and enter "" as the value.

Click **Select To**, expand **ValidationResponse > MainMessage** by clicking the '+' symbol, and click **MainMessage**.

Click **Add**. Click **Select From** and select **False** from the popup. Click **Select To**, expand **ValidationResponse > Validated** by clicking the '+' symbol, and click **Validated**.

Assign From	⇒	Assign To
""	⇒	ValidationResponse MainMessage xy
False	⇒	ValidationResponse Validated xy

This will provide default values for the **ValidationResponse** to pass to the Decision Service.

Select the **Assign1** activity and go to the Details tab of the Properties view. Click **Select From**, expand **DataValidationExecutionResult > DataValidationResponse** by clicking the '+' symbol, and click **DataValidationResponse**. Click **Select To** and click **ValidationResponse**.

Assign From	⇒	Assign To
DataValidationExecutionResult DataValidationResponse xy	⇒	ValidationResponse

This will extract the **ValidationResponse** result returned by the Decision Service.

The same sequence of steps need to be undertaken for the Eligibility_AIS_Process.

Double-click **Insurance_Sample_Implementation > Integration Logic > BPEL Processes > Eligibility_AIS_Process** and click the '+' symbol on the Reference Partners palette to add a new Reference Partner. Enter **AutoQuoteDecisionServicePartner** as the name and choose **AutoQuoteDecisionService** from Matching interfaces, then click OK.

Click **Basic Actions > Invoke** from the palette and click between the **Receive** and **Reply** activities.

Select the Invoke activity and go to the Details tab of the Properties view. Click **Browse...** next to the Partner field and select **AutoQuoteDecisionServicePartner**. Select **Eligibility** for the Operation.

Further down, click on **(none)** under Read from Variable next to AutoQuoteRequest and select **AutoQuoteRequest** in the popup, and click on **(none)** next to EligibilityResponse and select **EligibilityResponse** from the popup.

Under Store into Variable, click on **(none)** and select **New...** Click **OK** in the dialog that appears.

Invoke - Invoke

Description	Partner:*	AutoQuoteDecisionServicePartner	Browse...
Details	Interface:*	AutoQuoteDecisionService	
Server	Operation:*	Eligibility	
Administration	<input checked="" type="checkbox"/> Use data type variables mapping		
Exit Condition			
Compensation			
Correlation			
Environment			
Event Monitor			
Global Event Settings			

	Name	Type	Read from Variable	
Inputs	AutoQuoteRequest	AutoQuoteRequest	AutoQuoteRequest	⇒
	EligibilityResponse	EligibilityResponse	EligibilityResponse	⇒
	Name	Type	Store into Variable	
Outputs	EligibilityExecutionResult	EligibilityExecutionResult	⇒	EligibilityExecutionResult

In the process editor, select **Basic Actions > Assign** from the palette and click between the **Receive** and **Invoke** activities.

Select **Basic Actions > Assign** from the palette and click between the **Invoke** and **Reply** activities.

Select the **Assign** activity and go to the Details tab of the Properties view.

Click **Select From**, select **String (enter a value)** from the popup, and enter "" as the value. Click **Select To**, expand **EligibilityResponse > MainMessage** by clicking the '+' symbol, and click **MainMessage**.

Click **Add**. Click **Select From** and select **True** from the popup. Click **Select To**, expand **EligibilityResponse > Eligible** by clicking the '+' symbol, and click **Eligible**.

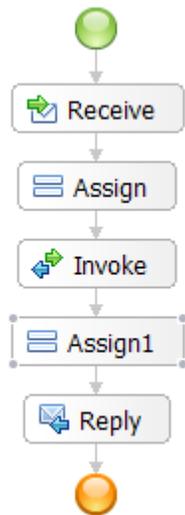
Assign From	⇒	Assign To
""	⇒	EligibilityResponse MainMessage
True	⇒	EligibilityResponse Eligible

Select the **Assign1** activity and go to the Details tab of the Properties view.

Click **Select From**, expand **EligibilityExecutionResult > EligibilityResponse** by clicking the '+' symbol, and click **EligibilityResponse**. Click **Select To** and click **EligibilityResponse**.

Assign From	⇒	Assign To
EligibilityExecutionResult EligibilityResponse	⇒	EligibilityResponse

The resulting process flow looks like this.

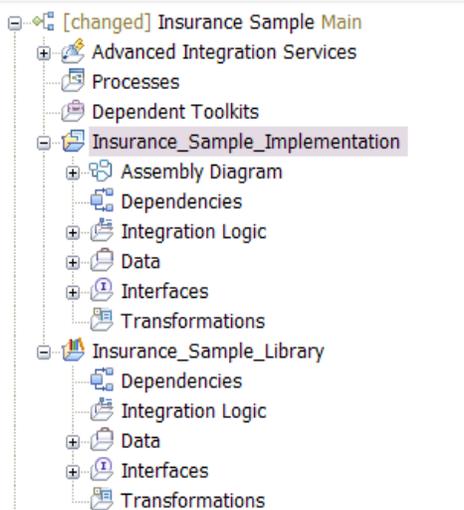


Click the **Save** button and all errors should be resolved.

Step 4. Defining the AIS Module Deployment Descriptor.

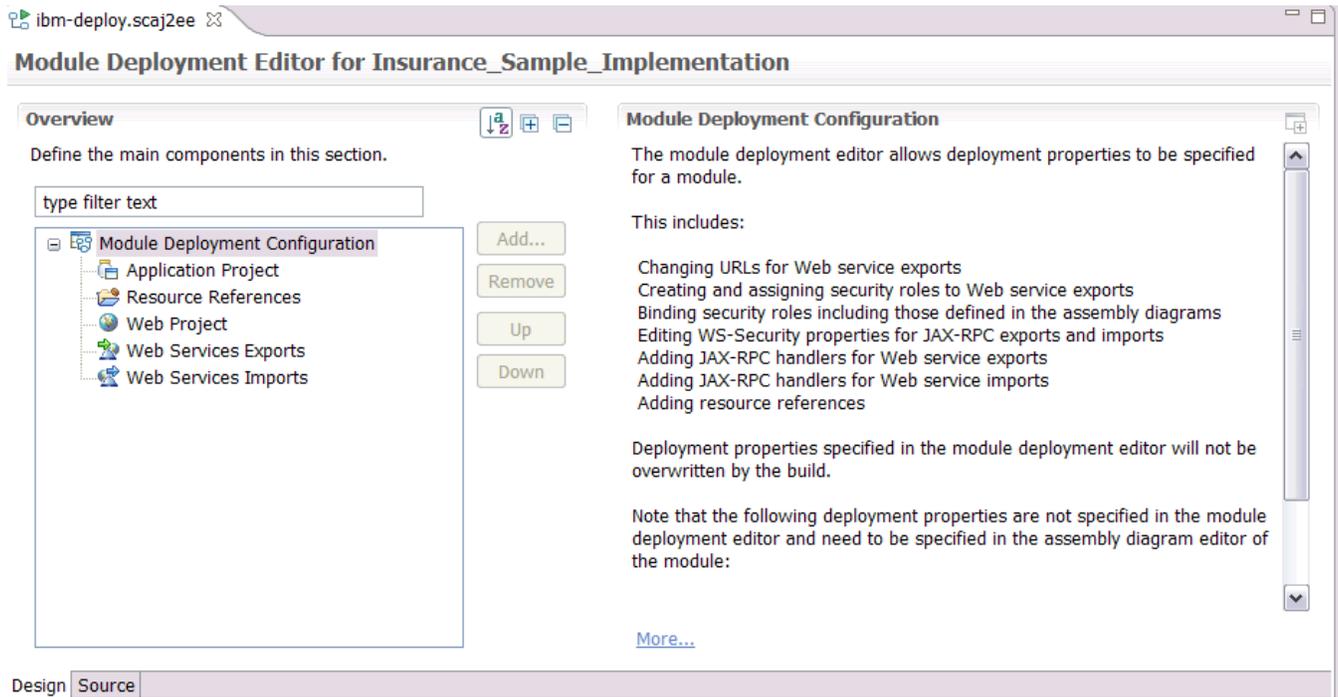
When working with AIS modules, the approach to defining the deployment descriptor is different. With previous tasks the deployment descriptor has been defined in the Web Application containing the module. With an AIS module, there is no containing module so the deployment descriptor has to be set up manually.

To setup the deployment descriptor
 Select the **Insurance_Sample_Implementation** module.

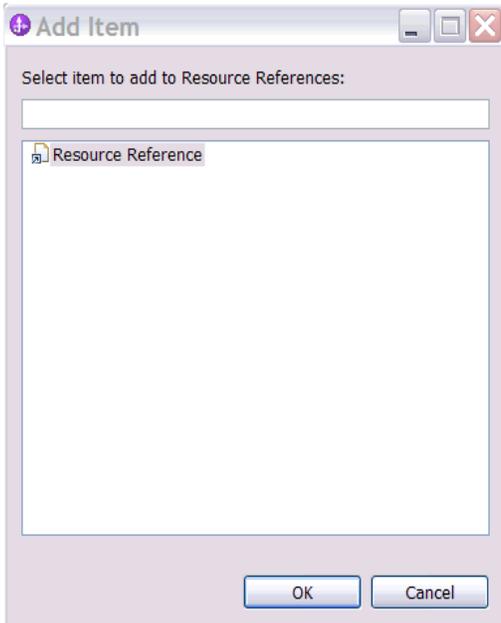


Right click and select **Open Deployment Editor**

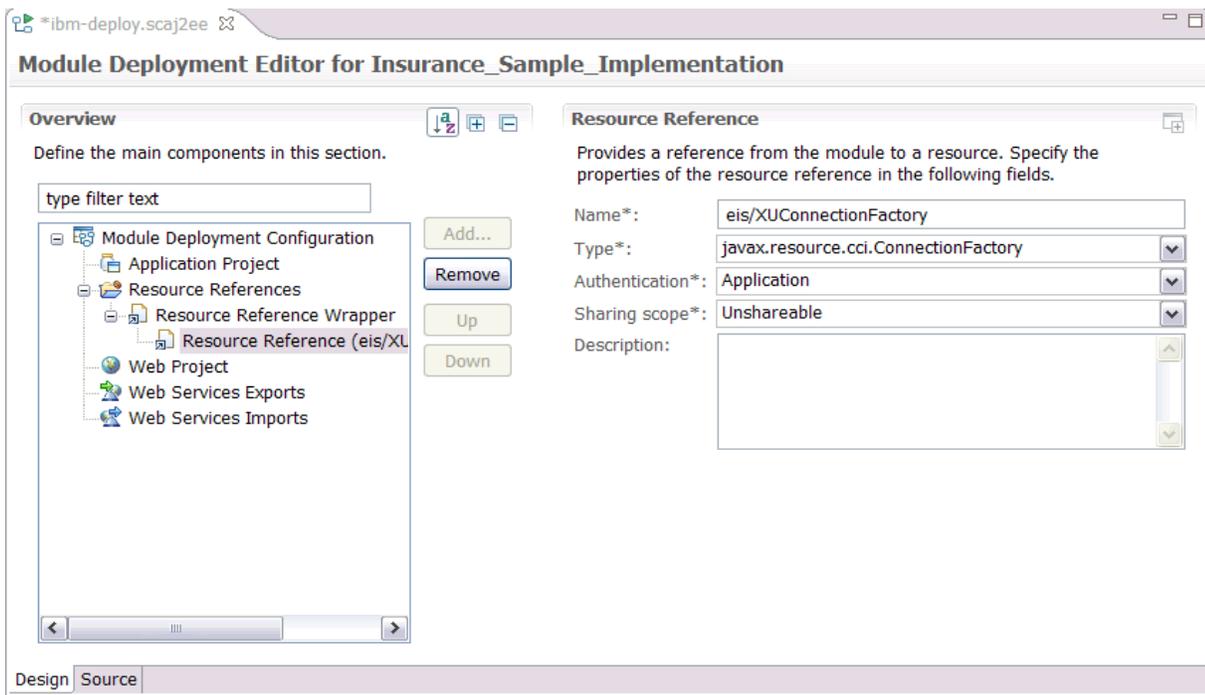
This will open up a deployment editor for the SCA Module as shown below.



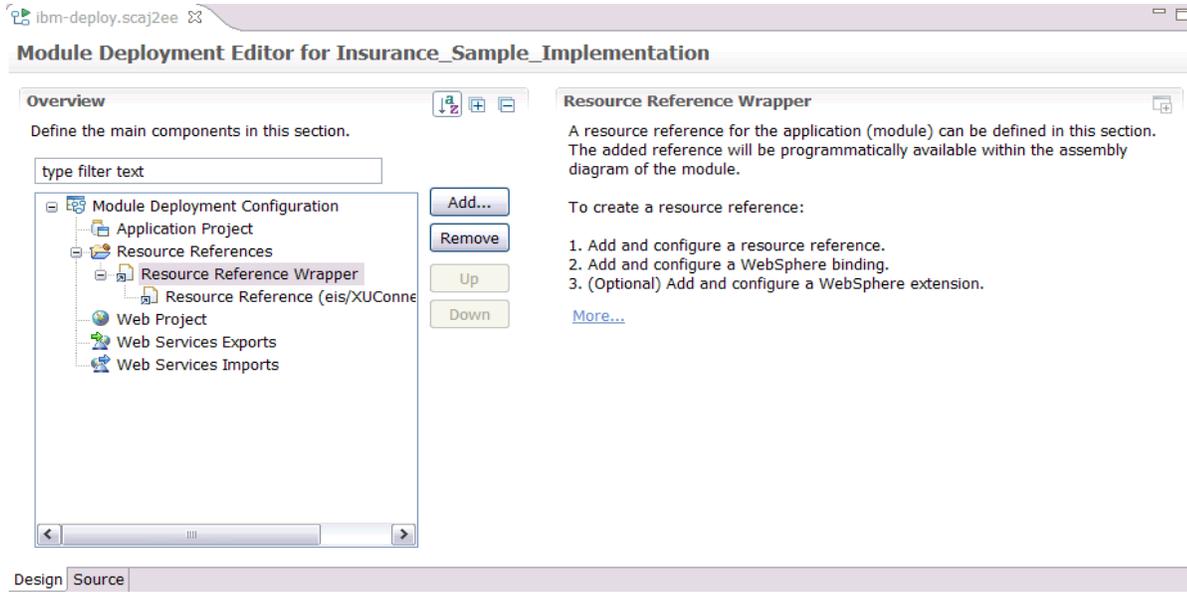
Select **Resource References** and click **Add...**



In the Add Item popup select the **Resource Reference** and Click **OK**.
Select the resource reference you have created and define the fields for the XU as shown below.

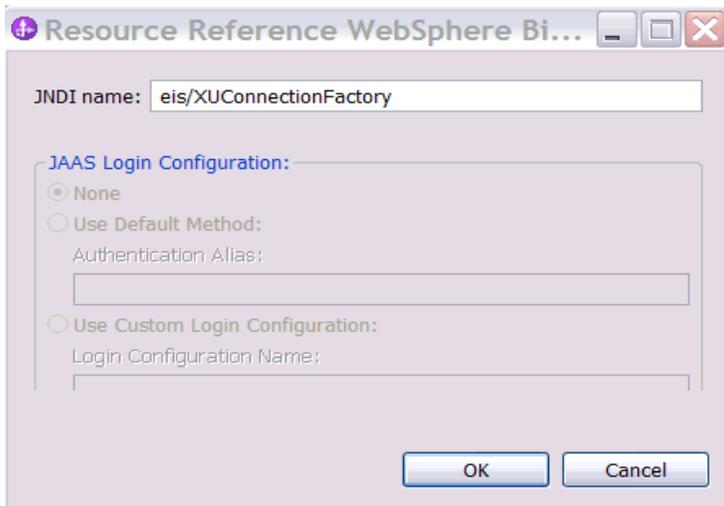


Save your work and select the Resource Reference Wrapper as shown below.

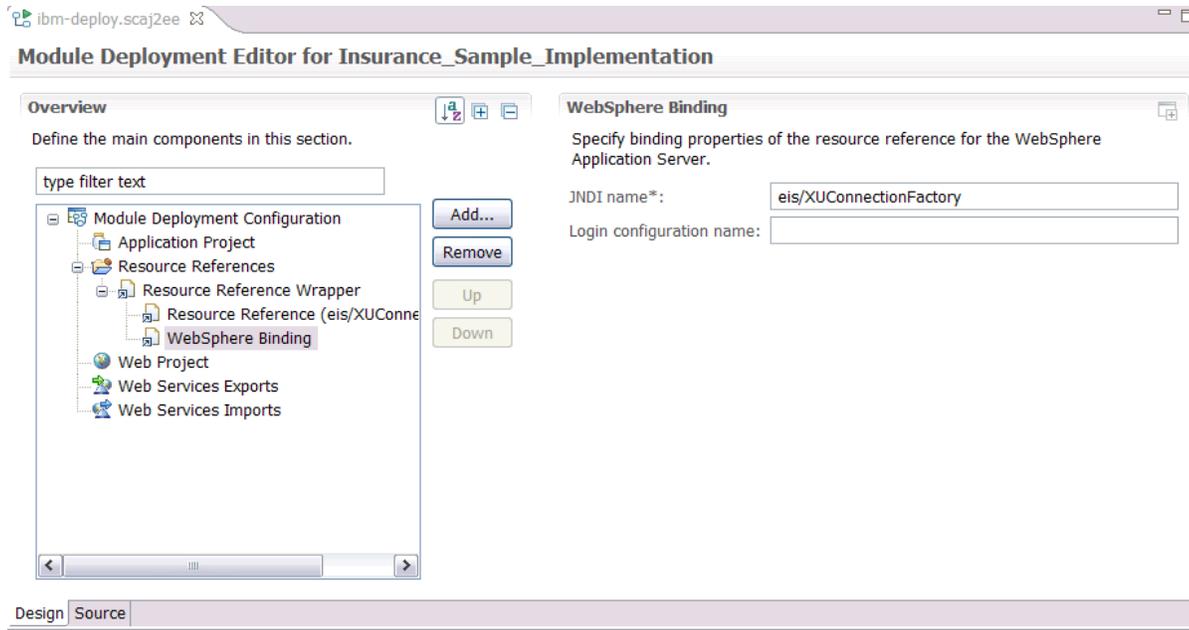


To create the WebSphere binding right click the Resource Reference Wrapper and Select **Add > WebSphere Binding**.

In the Resource Reference WebSphere Binding popup enter the JNDI name as shown below.



Click **OK**.



You have now configured the SCA module to refer to the eXecution Unit deployed on the same server as the module itself.

Step 5. Publishing back to Process Center and testing.

Before this implementation can be used it needs to be published back to Process Center. Select the **Insurance Sample Main** project. Right Click and select **Refresh and Publish**.

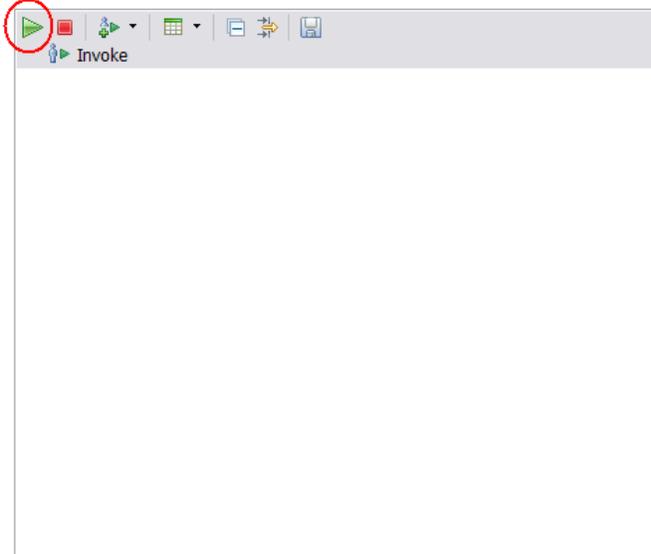
Open the **Assembly Diagram** select the **Eligibility_AIS** interface. Right click and select **Test Component**.

In the **Integration Test Client Value editor** set the **NumberOfAccidents** to **5**. Click the **Continue** icon.

Integration Test Client: Insurance_Sample_Implementation_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, and then click the Continue icon in the Events area to run the test. [More...](#)

Configuration: Default Module Test

Module: Insurance Sample/Insurance_Sample_Implementation

Component: Eligibility_AIS

Interface: Eligibility_AIS

Operation: invoke

Binding type is supported ?

Initial request parameters:

Value editor XML editor

Name	Type	Value
FirstDrivingLicenseDt	dateTime	2011-06-16T14...
VehicleVandalizedOrStol	boolean	false
LicenseSuspendedOrRev	boolean	false
DUI	boolean	false
NumberOfAccidents	int	5

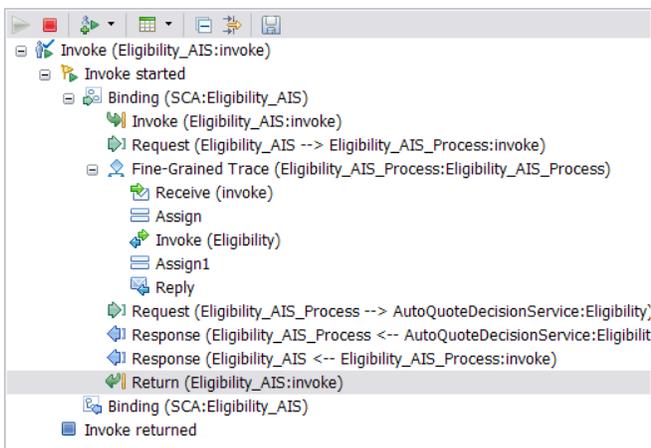
In the **Select a Deployment Location** select your **IBM Process Center** server.
 Ensure the user login credentials are correct for the Process Center
 For **User ID**: type **admin**
 For **Password**: type **admin**
 Click **OK**.

The trace of the **Eligibility_EIS** microflow should be clearly visible and the response from the Managed Decision service visible in the **EligibilityResponse**.

Integration Test Client: Insurance_Sample_Implementation_Test

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



General Properties

Detailed Properties

Process Application: Insurance Sample
 Module: [Insurance_Sample_Implementation](#)
 Component: [Eligibility_AIS](#)
 Interface: [Eligibility_AIS](#)
 Operation: [invoke](#)

Return parameters:

Value Editor XML Source

Name	Type	Value
EligibilityRe...	Eligibility...	
Eligible	boolean	false
MainMess	string	The driver has had too many accidents
ErrorMess	string[]	

This completes the implementation and testing of the **Managed Decision** Advanced Implementation Service.

A Process Center Repository snapshot is provide in
[SupportPac LA71 Path]\BPMTutorial\task7\Insurance_Sample - LA71TT7.twx

You can create your own snapshot or load this completed snapshot before continuing to task 8.