



SupportPac LA71: IBM WebSphere Operational Decision Management 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

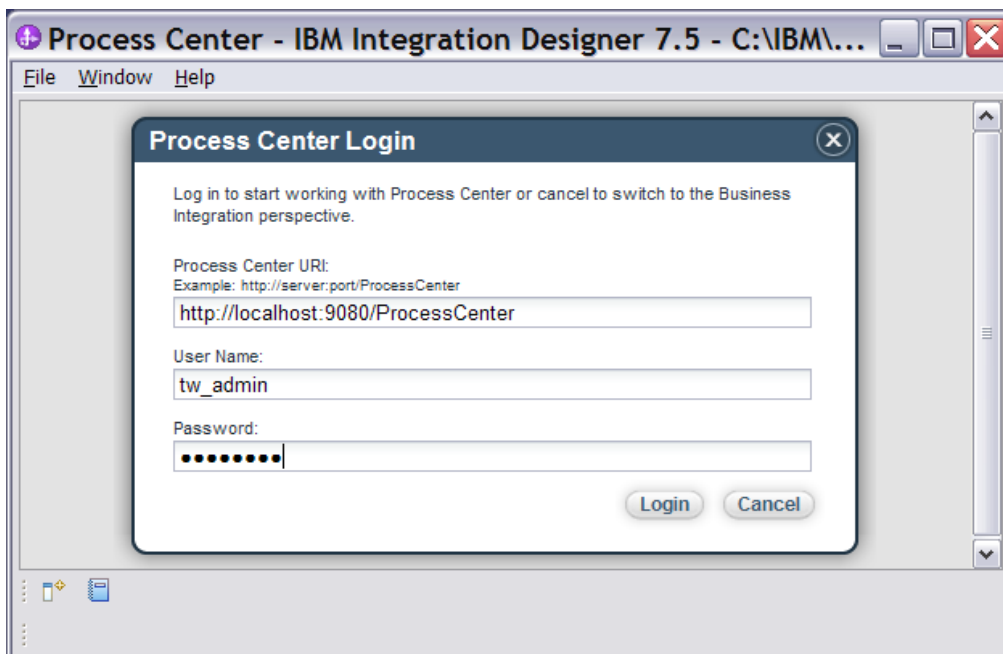
You should then open the Process App in the IID workspace and test the integration as described in step 5.

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

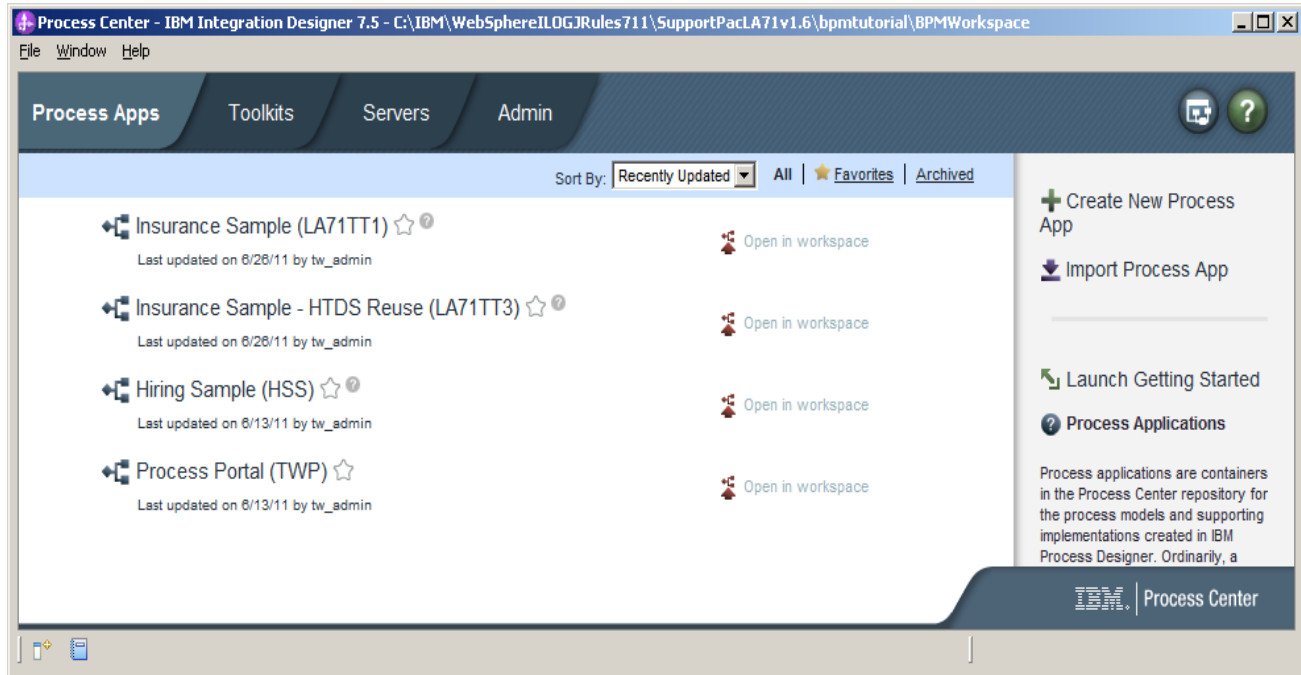
In Task 1 the Business Process Author designed the Decisions that were required and specified Advanced Integration Services interfaces that should be used to invoke the Managed Decision services that IT should develop. In this step you will bring the definition of those interfaces into your workspace.

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

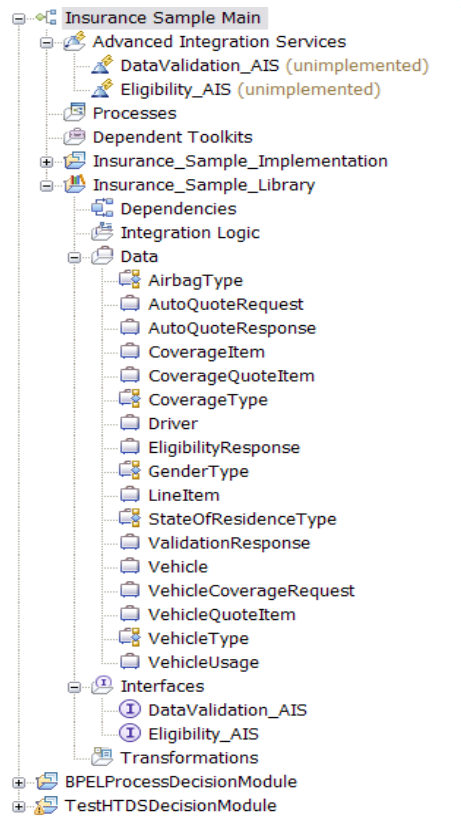
At the Process Center login enter the **Process Center URI** to correspond to your environment. Enter the User Name and Password – use tw_admin / tw_admin for a default install. Click **Login**.



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 will need to switch to the Business Integration Detailed View.

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.

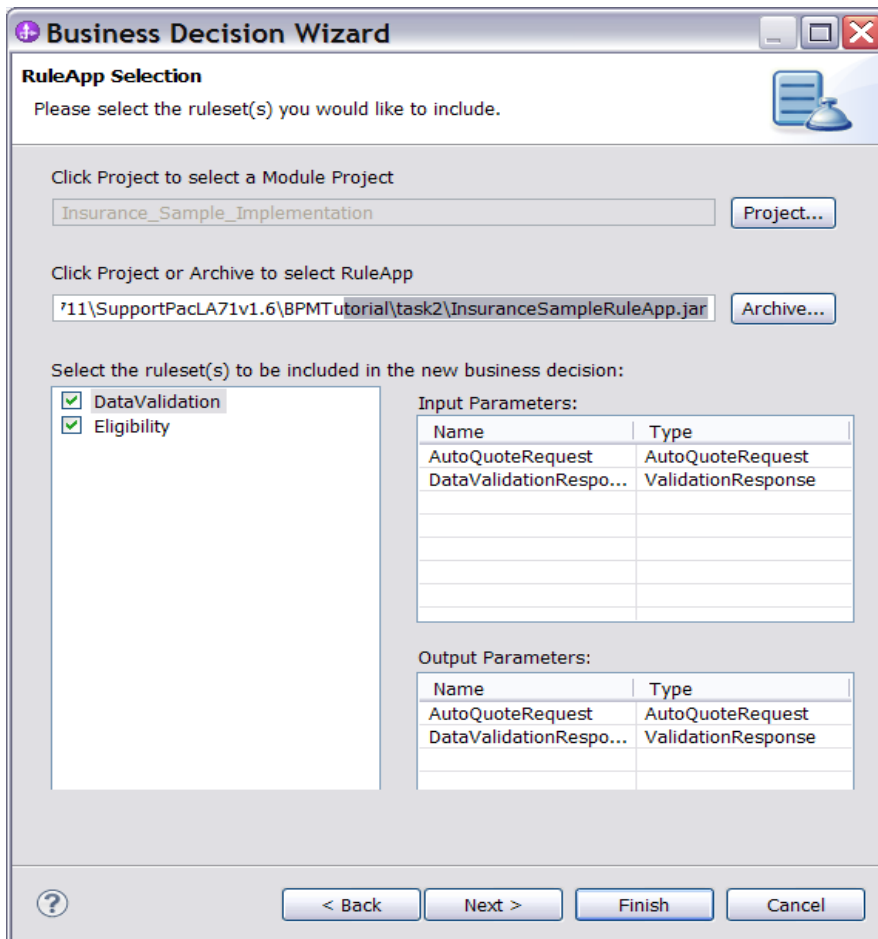
Right click and select **SupportPac LA71 > Create SCA Component from RuleApp**

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...**

Navigate to the directory where you saved the RuleApp archive (or to the completed BPMTutorial/task2 directory) and select **InsuranceSampleRuleApp.jar**



The screenshot shows the 'Business Decision Wizard' window, specifically the 'RuleApp Selection' step. The window has a title bar with a question mark icon and standard window controls. The main content area is titled 'RuleApp Selection' and contains the instruction 'Please select the ruleset(s) you would like to include.' Below this, there are two sections for selection:

- Click Project to select a Module Project:** A text field contains 'Insurance_Sample_Implementation' and a 'Project...' button is to its right.
- Click Project or Archive to select RuleApp:** A text field contains the file path 'I:\SupportPacLA71v1.6\BPMTutorial\task2\InsuranceSampleRuleApp.jar' and an 'Archive...' button is to its right.

Below these sections, there is a list of rulesets to be included in the new business decision:

- ☒ DataValidation
- ☒ Eligibility

To the right of the ruleset list, there are two tables for input and output parameters:

Input Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationRespo...	ValidationResponse

Output Parameters:

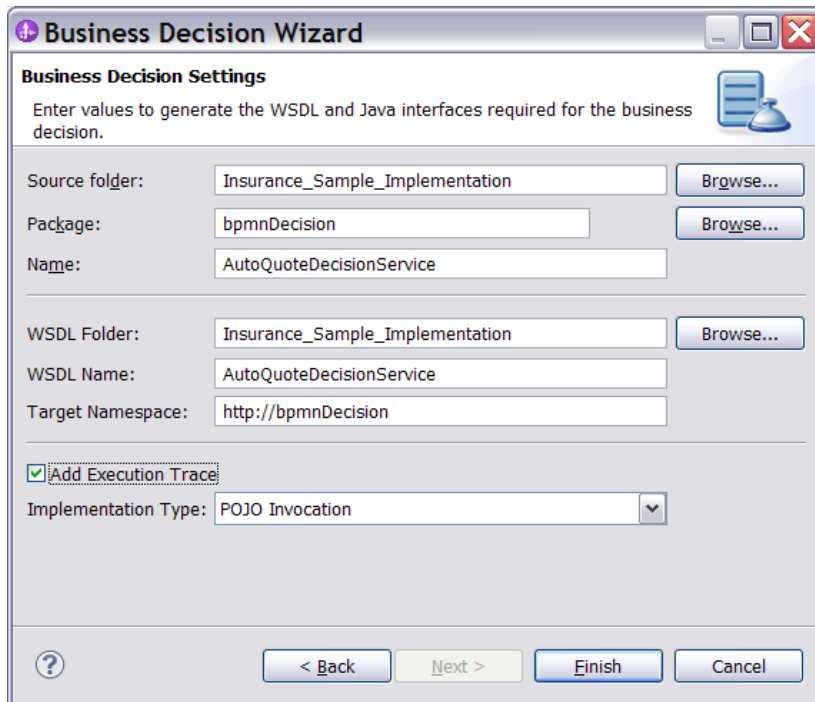
Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationRespo...	ValidationResponse

At the bottom of the wizard, there is a navigation bar with a question mark icon, a '< Back' button, a 'Next >' button, a 'Finish' button, and a 'Cancel' button.

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**.



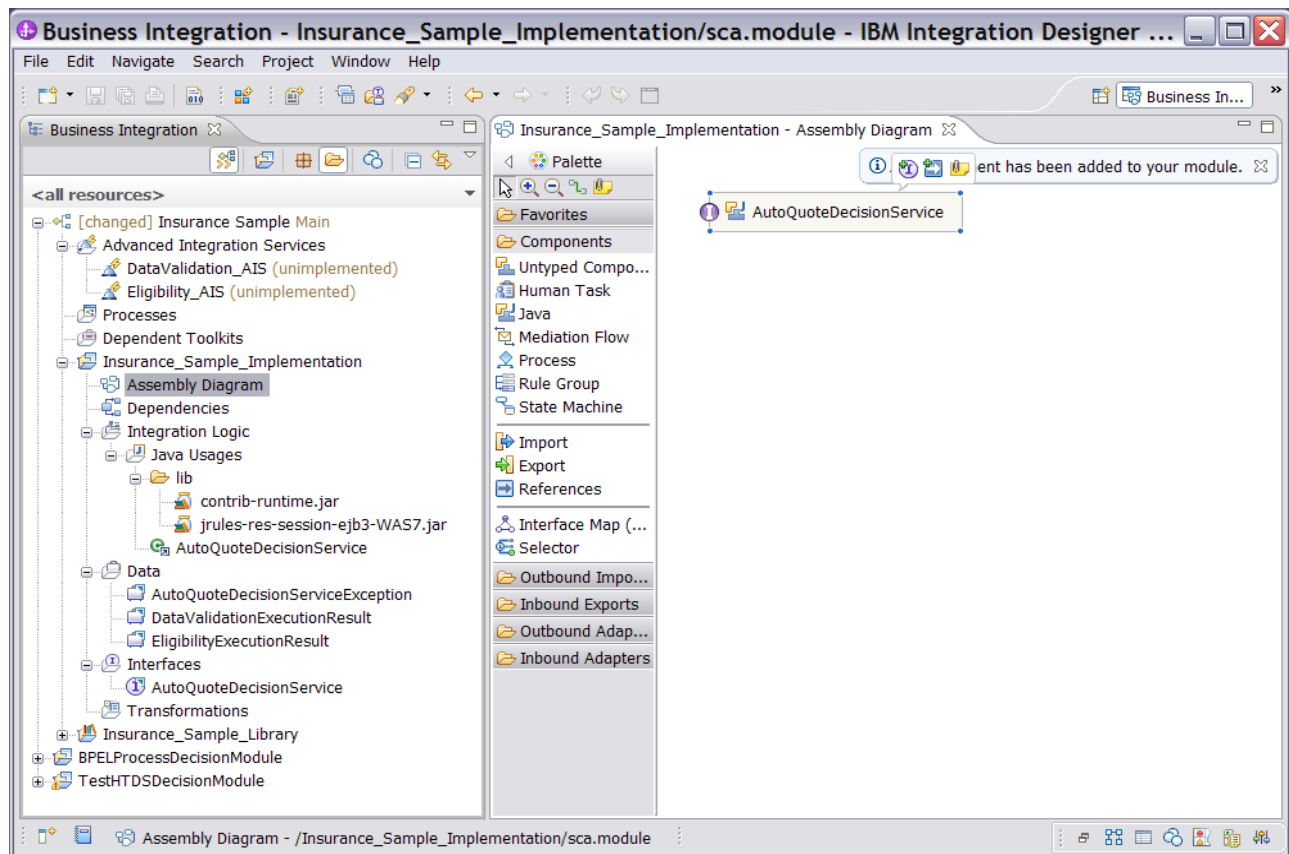
The screenshot shows the 'Business Decision Wizard' window with the 'Business Decision Settings' tab selected. The window title is 'Business Decision Wizard'. Below the title bar, there's a subtitle 'Business Decision Settings' and a description: 'Enter values to generate the WSDL and Java interfaces required for the business decision.' To the right of the description is a small icon of a document with a bell. The form contains several input fields and buttons:

- Source folder:** A text box containing 'Insurance_Sample_Implementation' and a 'Browse...' button.
- Package:** A text box containing 'bpmnDecision' and a 'Browse...' button.
- Name:** A text box containing 'AutoQuoteDecisionService'.
- WSDL Folder:** A text box containing 'Insurance_Sample_Implementation' and a 'Browse...' button.
- WSDL Name:** A text box containing 'AutoQuoteDecisionService'.
- Target Namespace:** A text box containing 'http://bpmnDecision'.
- Add Execution Trace:** A checkbox that is checked.
- Implementation Type:** A dropdown menu showing 'POJO Invocation'.

At the bottom of the window, there are four buttons: a help button (question mark icon), '< Back', 'Next >', 'Finish', and '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**.

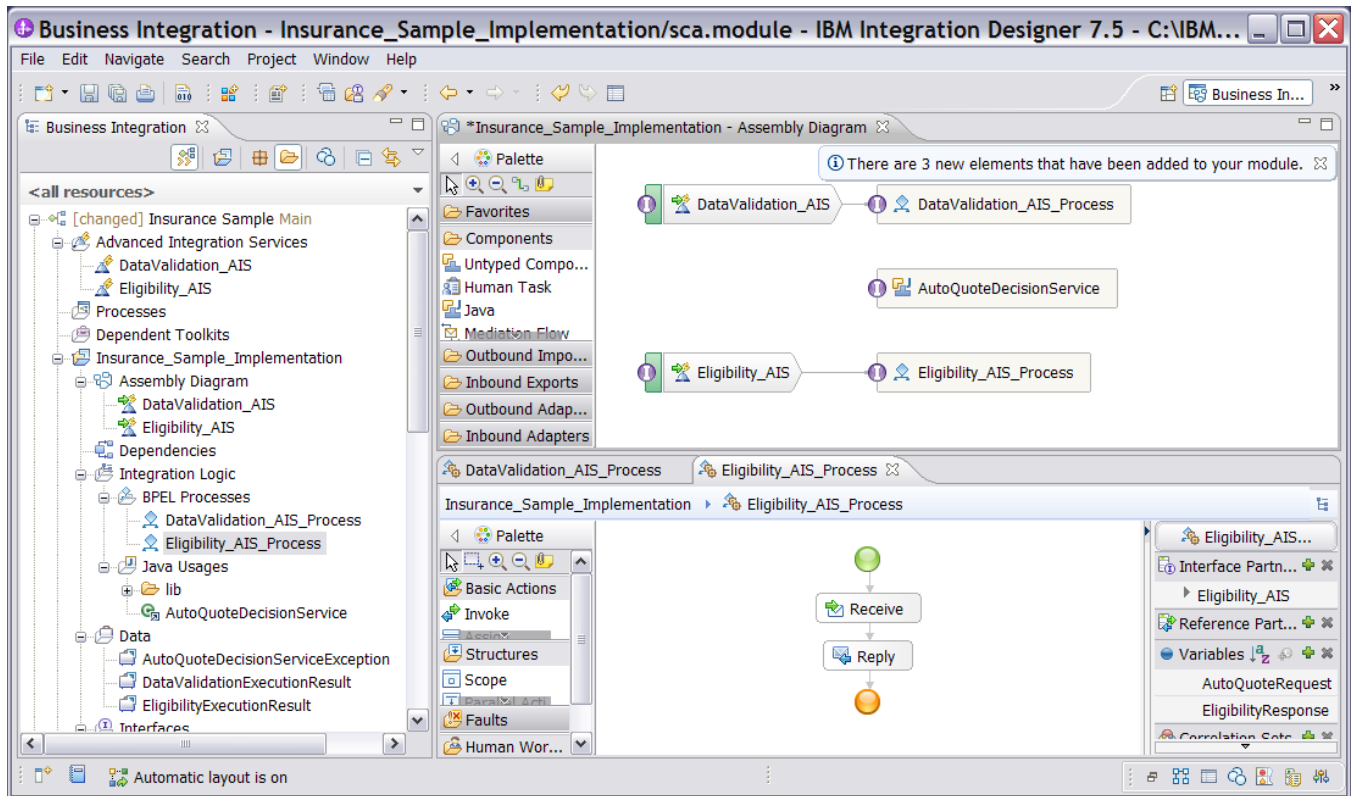
Select **Insurance Sample > Advanced Integration Services > Eligibility_AIS**

Right Click and select **Implement**.

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

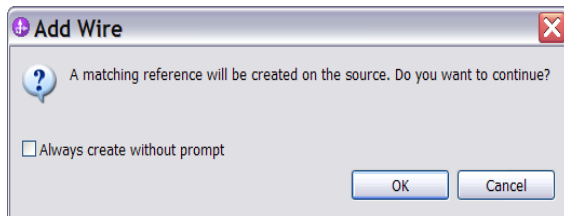
Click **Finish**.

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.

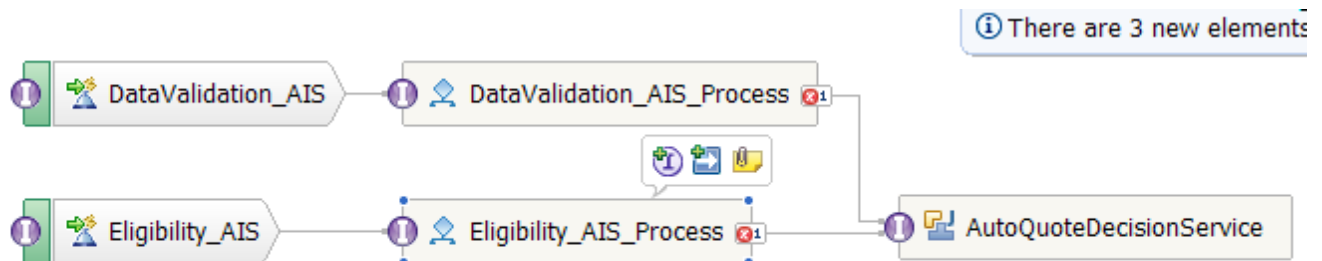


You now need to bind the microflows to the decision service.
Select the DataValidation_AIS_Process and drag a wire from it to the AutoQuoteDecisionService.

Click **OK** to the popup.

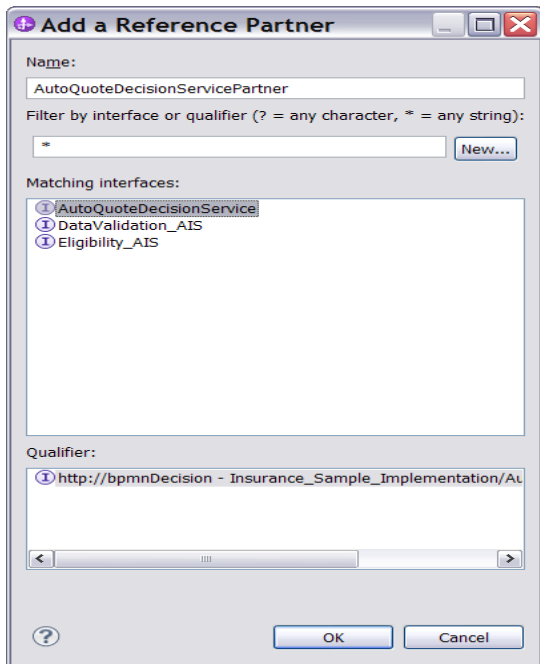


Select the **Eligibility_AIS_Process** and drag a wire from it to the **AutoQuoteDecisionService**.
Click **OK** to the popup.



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

Open the process editor for **DataValidation_AIS_Process**,
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 under the Receive element.
Select the Invoke Element and Properties screen in the Details tab.

For partner Click the **Browse...** button and select AutoQuoteDecisionPartner.
Ensure the interface is **AutoQuoteDecisionService**.
For Operation select **DataValidation**.

Fill in the Input Read from Variables from existing variables.
For the outputs create a new variable accepting the defaults.

Task Flows Properties Build Activities Problems Server Logs

Invoke - Invoke

Description	Partner:* AutoQuoteDecisionServicePartner Browse...
Details	Interface:* AutoQuoteDecisionService
Server	Operation:* DataValidation
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
	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.
 Select **Basic Actions > Assign** and click after **Receive**.
 Select **Basic Actions > Assign** and click after **Invoke**.

The Process Editor looks like this now.






Select the **Assign** Node, and in the Properties screen, Details tab initialize **ValidationResponse**.

Assign From	Assign To
""	ValidationResponse MainMessage
False	ValidationResponse Validated

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

Select the Assign1 Node, and in the Properties screen, Details tab extract the **ValidationResponse** from the **DataValidationExecutionResult**.

Assign From	Assign To
 DataValidationExecutionResult DataValidationResponse 	 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

Create the **AutoQuoteDecisionServicePartner**.







Add the Invoke to the Process flow to invoke the **AutoQuoteDecisionService** creating the new variable **EligibilityExecutionResult**

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

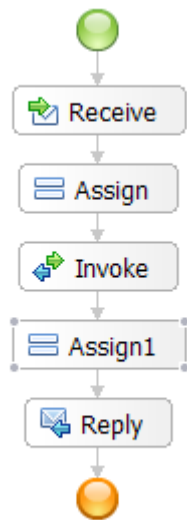
Add the Assign to initialize the **EligibilityResponse** variable.

Assign From	Assign To
 ""	 EligibilityResponse MainMessage 
 True	 EligibilityResponse Eligible 

Add the Assign1 to extract the EligibilityResponse from the EligibilityExecutionResult.

Assign From	Assign To
 EligibilityExecutionResult EligibilityResponse 	 EligibilityResponse

The resulting process flow looks like this.

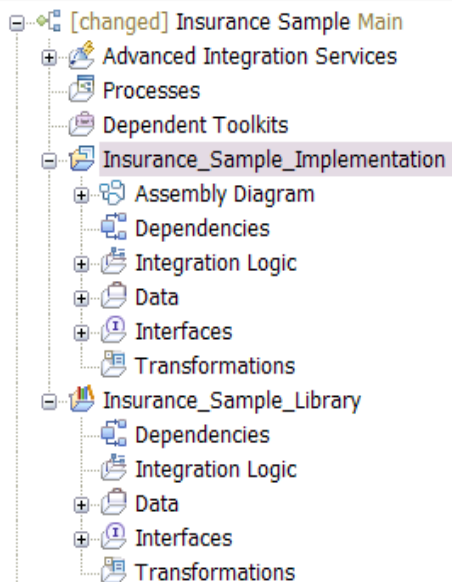


Clean and rebuild the projects and all errors should be removed.

Step 4. Defining the AIS Module Deployment Descriptor.

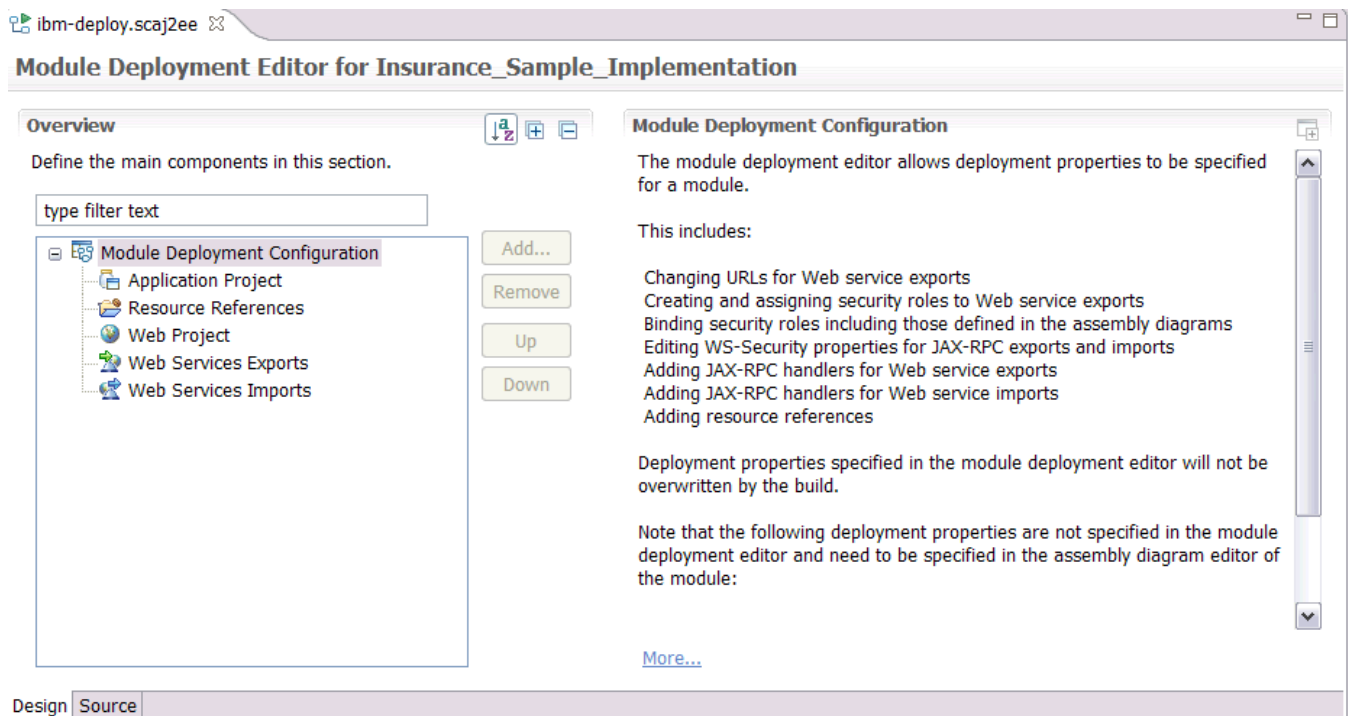
When working with AIS modules, the approach to defining the deployment descriptor is different. With previous tasks the deployment descriptor has been defined in the Web Application containing the module. With an AIS module, there is no containing module so the deployment descriptor has to be set up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

To setup the deployment descriptor
Select the Insurance_Sample_Implementation module.

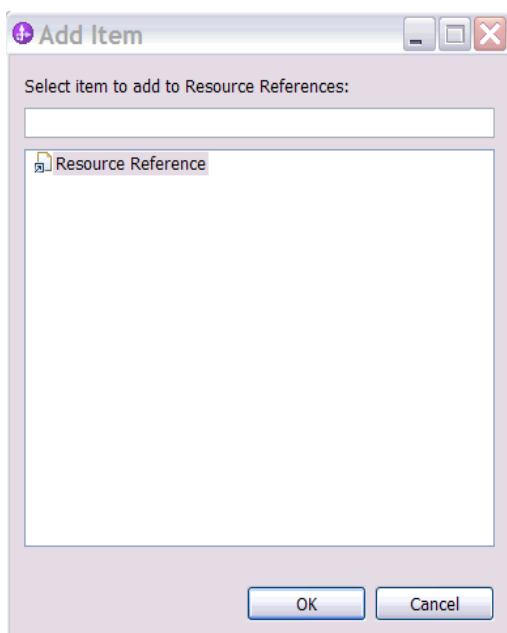


Right click and select Open Deployment Editor

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

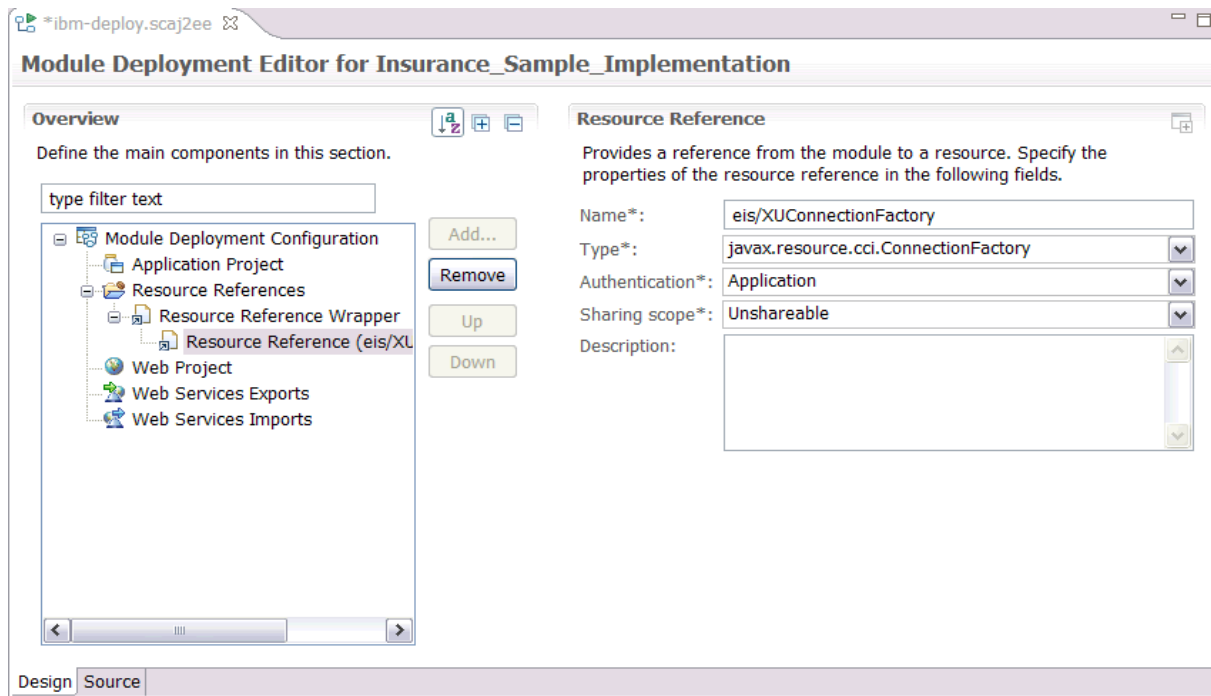


Select Resource References and click Add...

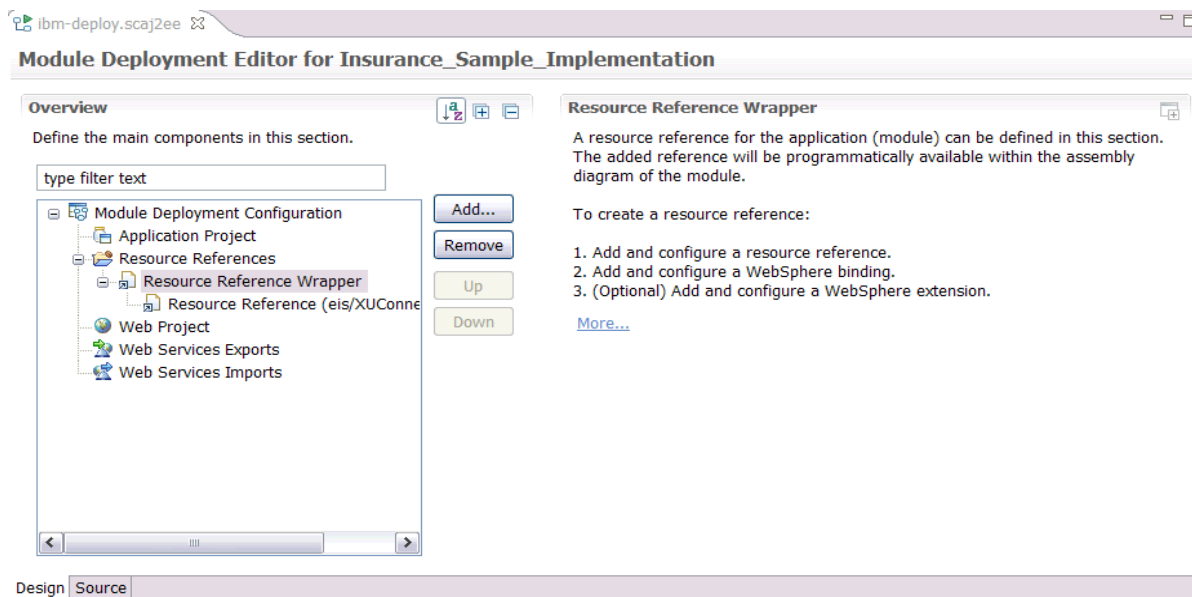


In the Add Item popup select the Resource Reference and Click OK.

Select the resource reference you have created and define the fields for the XU as shown below.

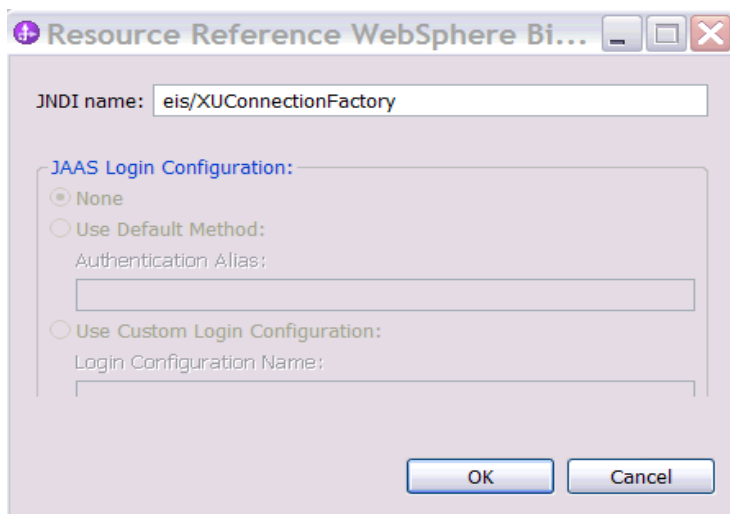


Save your work and select the Resource Reference Wrapper as shown below.

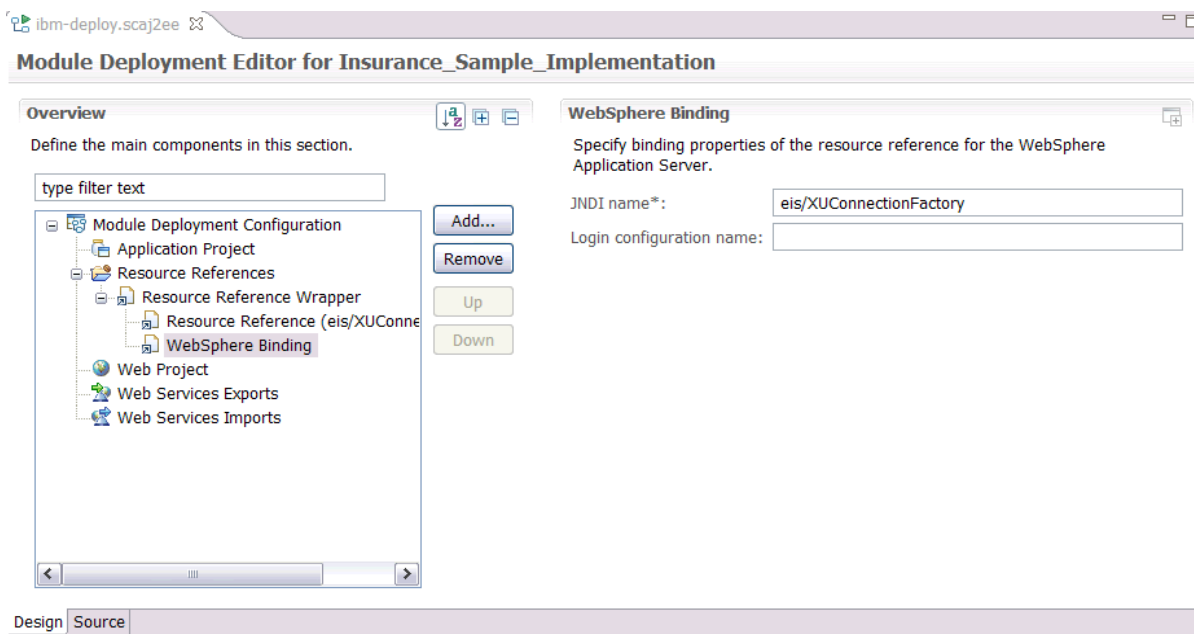


To create the WebSphere binding right click the Resource Reference Wrapper and Select Add > WebSphere Binding.

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 eXection 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** project. Right Click and select **Refresh and Publish**.

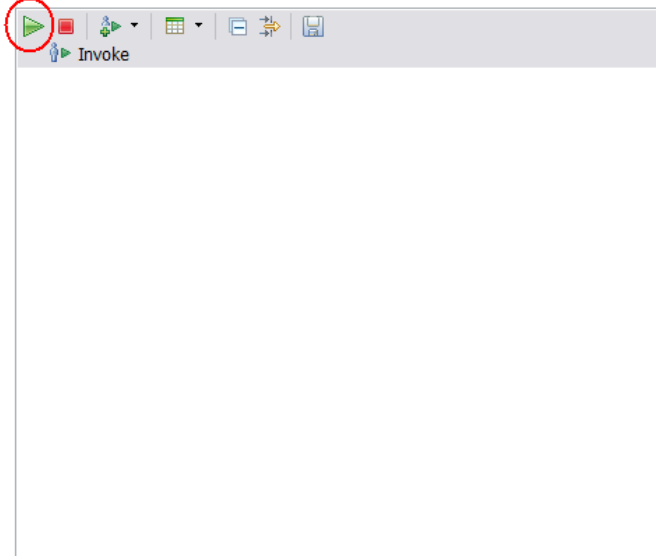
Open the **Assembly Diagram** select the **Eligibility_AIS** interface.
Right click and select **Test Component**.

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:...
VehicleVandalizedOrStole	boolean	false
LicenseSuspendedOrRevoked	boolean	false
DUI	boolean	false
NumberOfAccidents	int	5

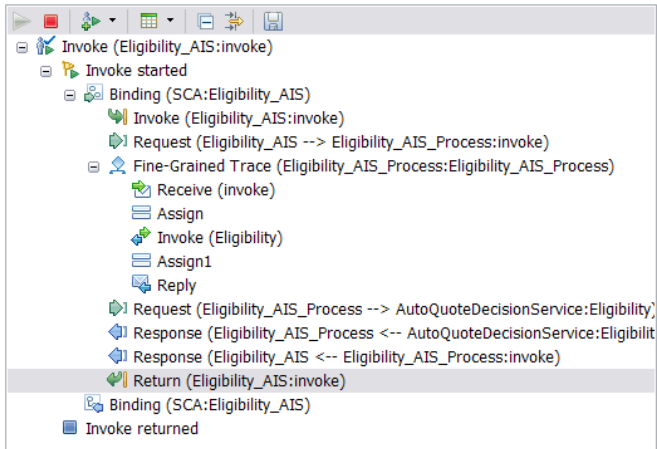
In the **Select a Deployment Location** select your **IBM Process Center** server.
Ensure the user login credentials are correct for the Process Center
For **User ID:** type **tw_admin**
For **Password:** type **tw_admin**
Click **OK**.

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...	ab
Eligible	boolean	ab false
MainMess	string	ab The driver has had too many accidents
ErrorMess	string[]	60

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

A Process Center Repository snapshot is provide in
SupportPacLA71v1.7\BPMTutorial\task7\Insurance_Sample – LA71TT7.twx

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