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

Trademarks

Overview

In this task you with 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.

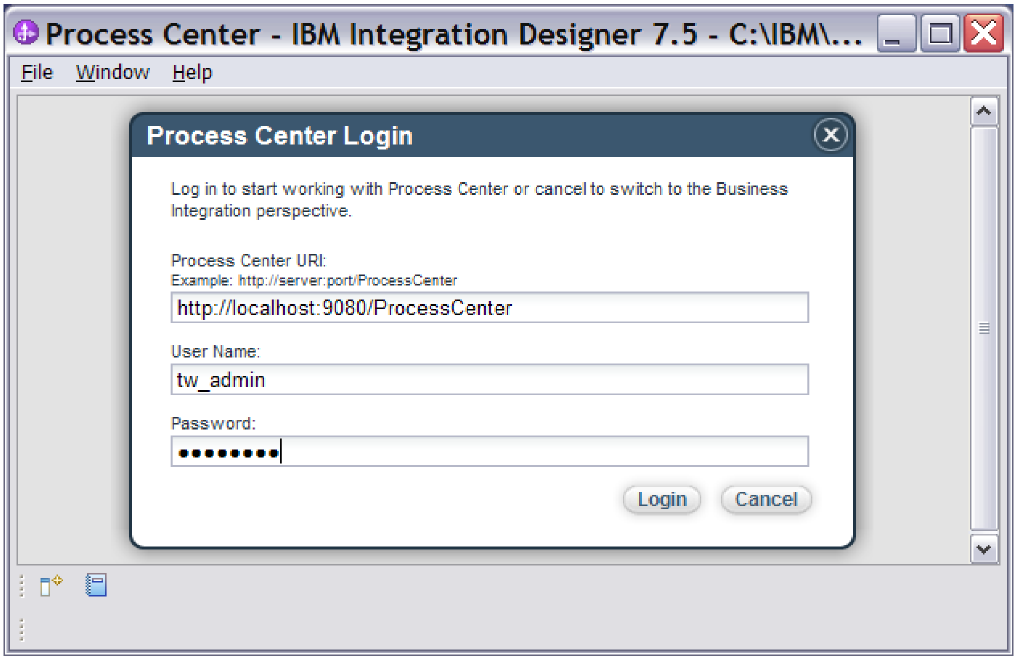
Readers who want to skip the exercise should import the task 7 solution into Process Center from the snapshot saved at
[SupportPac LA71 Path]\BPMTutorial\task7\Insurance_Sample - LA71TT7.twx
You should then open the Process App in the IID workspace and test the integration as described in step 5.

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**.



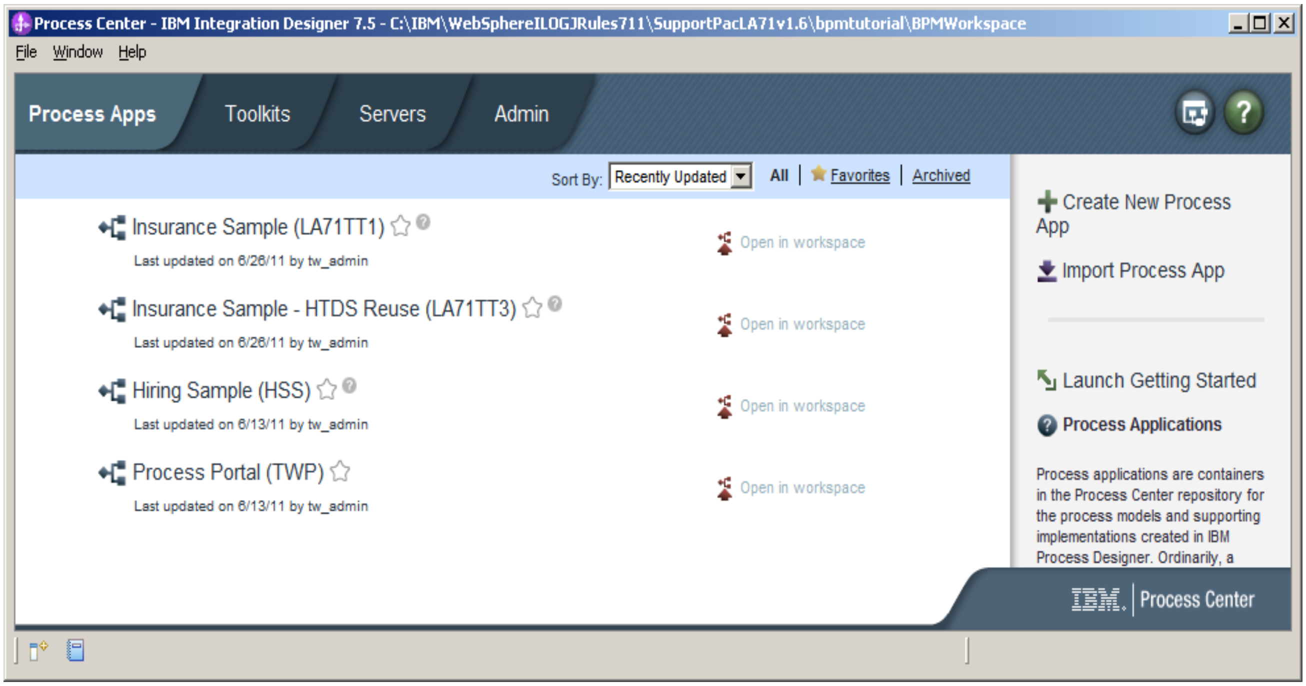
Log in to start working with Process Center or cancel to switch to the Business Integration perspective.

Process Center URI:
Example: http://server:port/ProcessCenter
http://localhost:9080/ProcessCenter

User Name:
tw_admin

Password:
••••••••

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


Process Center - IBM Integration Designer 7.5 - C:\IBM\WebSphereILOGJRules711\SupportPacLA71v1.6\bpmtutorial\BPMWorkspace
File Window Help
Process Apps
Toolkits
Servers
Admin
Sort By: Recently Updated
All | Favorites | Archived
Create New Process App
Import Process App
Launch Getting Started
Process Applications
IBM. Process Center

Insurance Sample (LA71TT1) — Last updated on 6/26/11 by tw_admin — Open in workspace

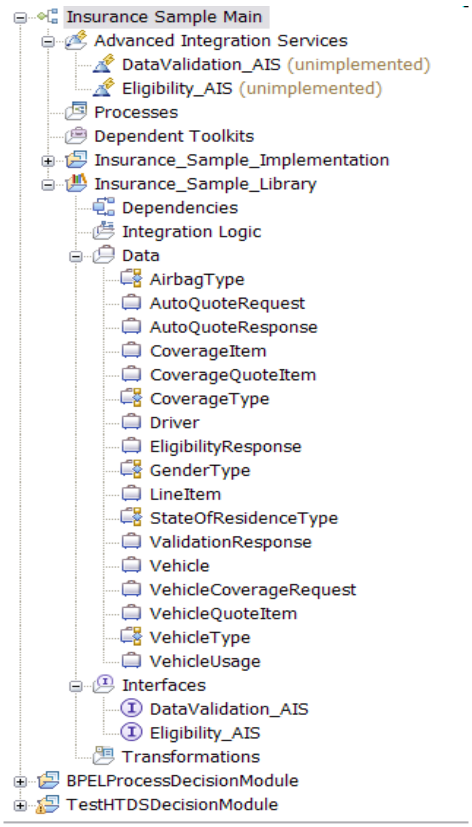
Insurance Sample - HTDS Reuse (LA71TT3) — Last updated on 6/26/11 by tw_admin — Open in workspace

Hiring Sample (HSS) — Last updated on 6/13/11 by tw_admin — Open in workspace

Process Portal (TWP) — Last updated on 6/13/11 by tw_admin — Open in workspace

Process applications are containers in the Process Center repository for the process models and supporting implementations created in IBM Process Designer. Ordinarily, a

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



- Insurance Sample Main
  - Advanced Integration Services
    - DataValidation_AIS (unimplemented)
    - Eligibility_AIS (unimplemented)
  - Processes
  - Dependent Toolkits
  - Insurance_Sample_Implementation
  - Insurance_Sample_Library
    - Dependencies
    - Integration Logic
    - Data
      - AirbagType
      - AutoQuoteRequest
      - AutoQuoteResponse
      - CoverageItem
      - CoverageQuoteItem
      - CoverageType
      - Driver
      - EligibilityResponse
      - GenderType
      - LineItem
      - StateOfResidenceType
      - ValidationResponse
      - Vehicle
      - VehicleCoverageRequest
      - VehicleQuoteItem
      - VehicleType
      - VehicleUsage
    - Interfaces
      - DataValidation_AIS
      - Eligibility_AIS
    - Transformations
- BPELProcessDecisionModule
- TestHTDSDecisionModule

If the Insurance_Sample_Implementation and Insurance_Sample_Library are not visible you will need to switch to the Business Integration Detailed View.

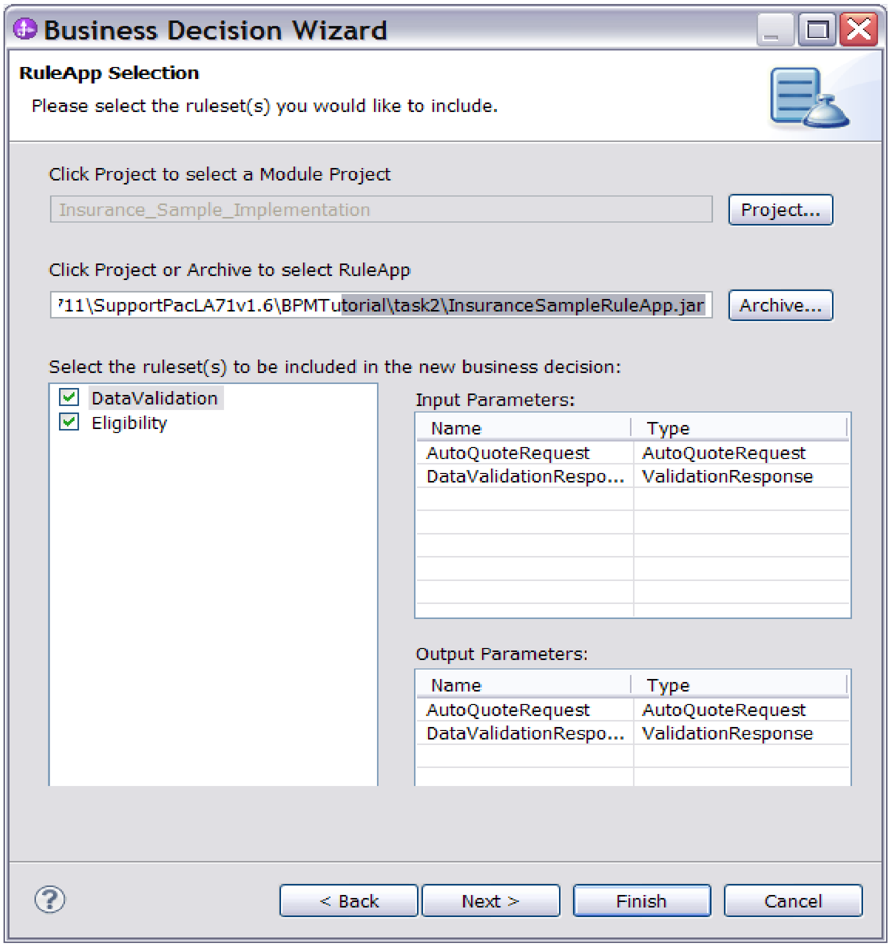
Note the two AIS interfaces (DataValidation_AIS and Eligibility_AIS) and the Business Objects that have been defined in the Insurance Sample Library. These Business Objects are the same as those used in the BRMS Decisions so we will not have any Data mapping to do.

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**



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

Click Project to select a Module Project

Insurance_Sample_Implementation

Project...

Click Project or Archive to select RuleApp

'11\SupportPacLA71v1.6\BPMTutorial\task2\InsuranceSampleRuleApp.jar

Archive...

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

☑ DataValidation
☑ Eligibility

Input Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationRespo...	ValidationResponse

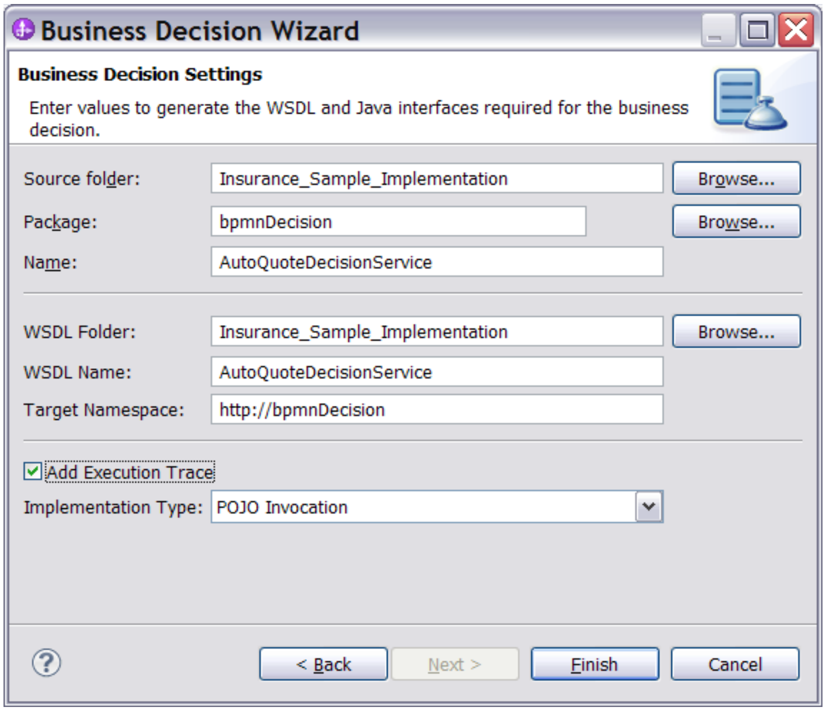
Output Parameters:

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**.



Business Decision Wizard

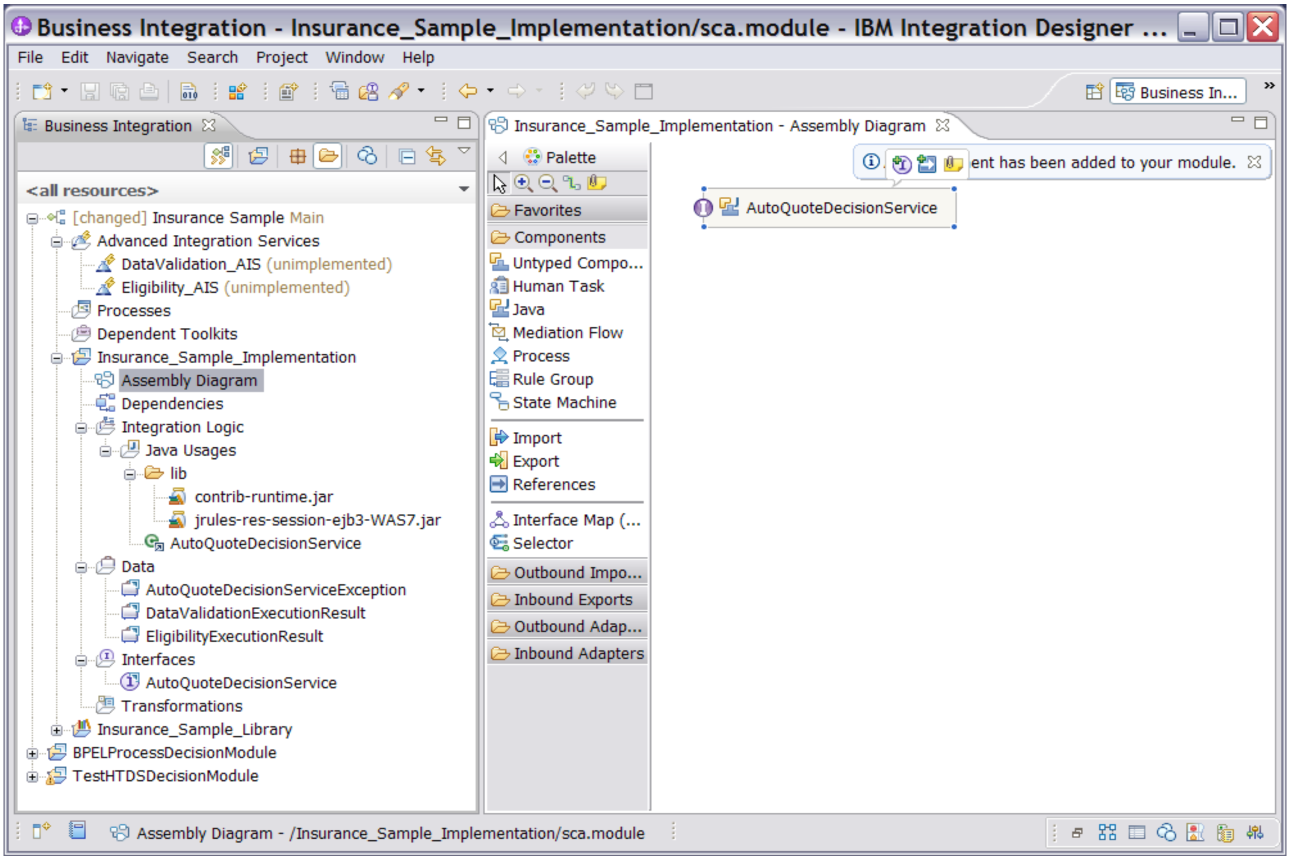
Business Decision Settings

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

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.


Business Integration - Insurance_Sample_Implementation/sca.module - IBM Integration Designer ...
File Edit Navigate Search Project Window Help
Business In...
Business Integration
<all resources>
[changed] Insurance Sample Main
Advanced Integration Services
DataValidation_AIS (unimplemented)
Eligibility_AIS (unimplemented)
Processes
Dependent Toolkits
Insurance_Sample_Implementation
Assembly Diagram
Dependencies
Integration Logic
Java Usages
lib
contrib-runtime.jar
jrules-res-session-ejb3-WAS7.jar
AutoQuoteDecisionService
Data
AutoQuoteDecisionServiceException
DataValidationExecutionResult
EligibilityExecutionResult
Interfaces
AutoQuoteDecisionService
Transformations
Insurance_Sample_Library
BPELProcessDecisionModule
TestHTDSDecisionModule
Insurance_Sample_Implementation - Assembly Diagram
Palette
Favorites
Components
Untyped Compo...
Human Task
Java
Mediation Flow
Process
Rule Group
State Machine
Import
Export
References
Interface Map (...
Selector
Outbound Impo...
Inbound Exports
Outbound Adap...
Inbound Adapters
AutoQuoteDecisionService
Assembly Diagram - /Insurance_Sample_Implementation/sca.module

ent has been added to your module.

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.

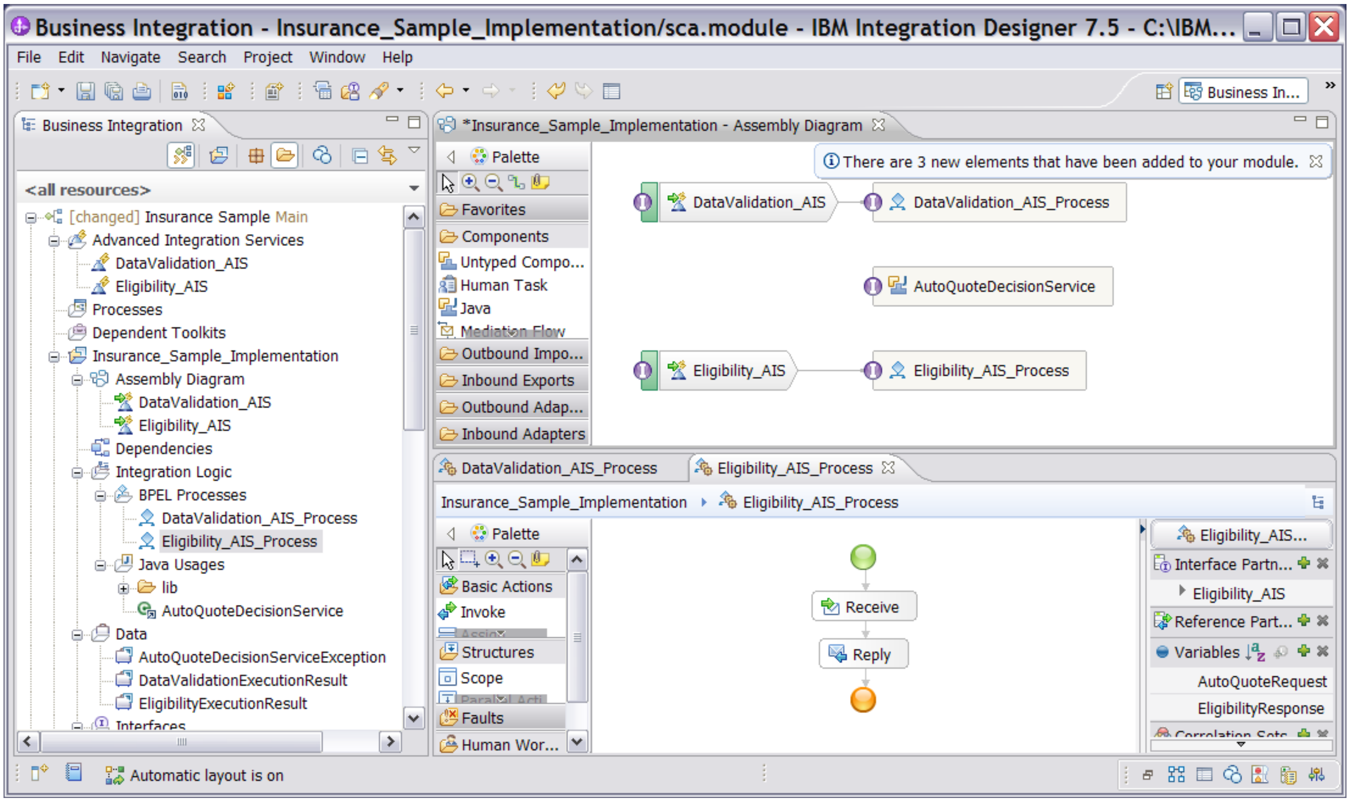
Step 3. Implement Advanced Integration Services.

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**.

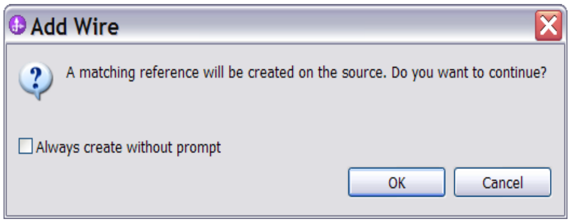
This creates microflow BPEL processes for each AIS and integrates them into the assembly diagram as shown below. You may need to right click in the Assembly Diagram canvas and select Automatic Layout to ensure the elements are laid out nicely.



There are 3 new elements that have been added to your module.

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.

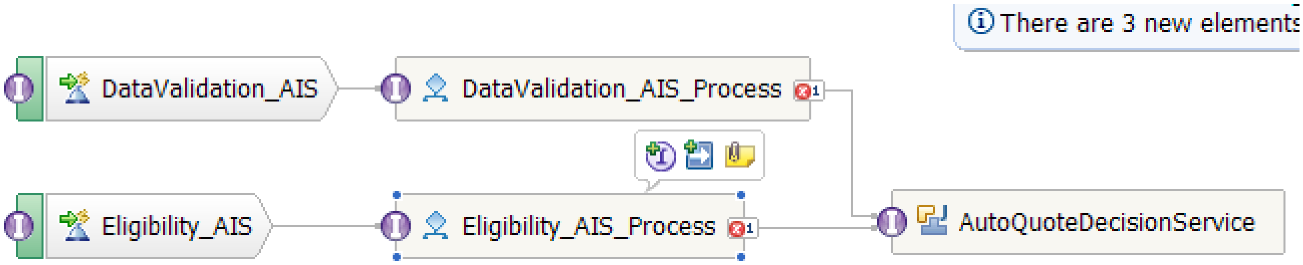
Click **OK** to the popup.



A matching reference will be created on the source. Do you want to continue?

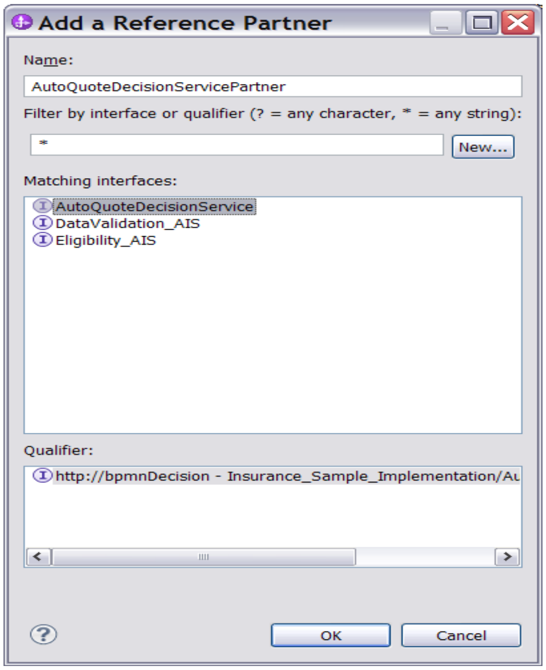
☐ Always create without prompt

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**.

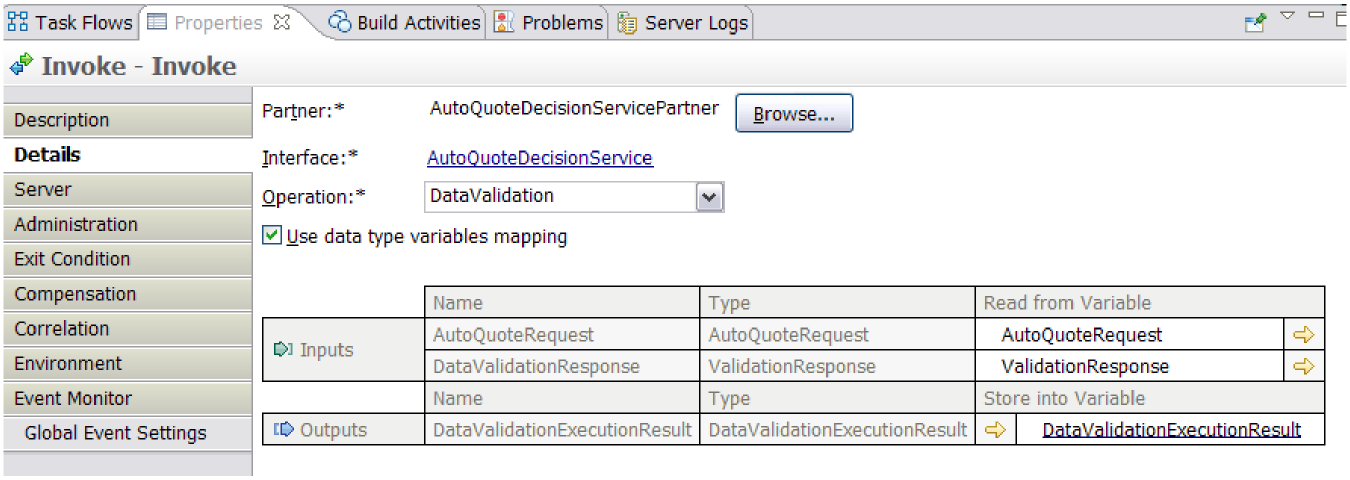


Name:
AutoQuoteDecisionServicePartner
Filter by interface or qualifier (? = any character, * = any string):
*
Matching interfaces:
AutoQuoteDecisionService
DataValidation_AIS
Eligibility_AIS
Qualifier:
http://bpmnDecision - Insurance_Sample_Implementation/Au

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.



Partner:* AutoQuoteDecisionServicePartner

Interface:* AutoQuoteDecisionService

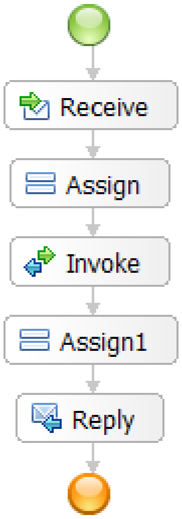
Operation:* DataValidation

☑ Use data type variables mapping

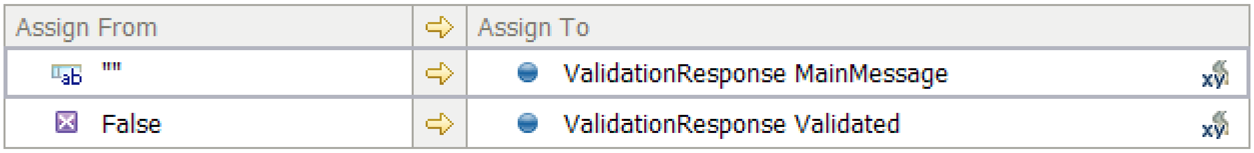
	Name	Type	Read from Variable
Inputs	AutoQuoteRequest	AutoQuoteRequest	AutoQuoteRequest
	DataValidationResponse	ValidationResponse	ValidationResponse
	Name	Type	Store into Variable
Outputs	DataValidationExecutionResult	DataValidationExecutionResult	DataValidationExecutionResult

Two last steps are needed to initialize the data for the request and map the response to the reply.
Select **Basic Actions > Assign** and click after **Receive**.
Select **Basic Actions > Assign** and click after **Invoke**.

The Process Editor looks like this now.



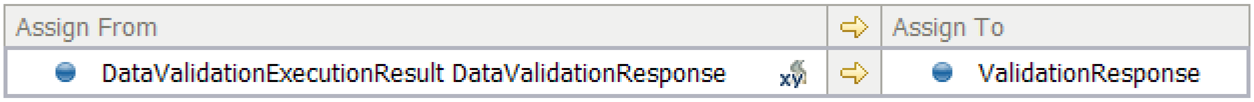
Select the **Assign** Node, and in the Properties screen, Details tab initialize **ValidationResponse**.



Assign From	⇨	Assign To
""	⇨	ValidationResponse MainMessage
False	⇨	ValidationResponse Validated

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

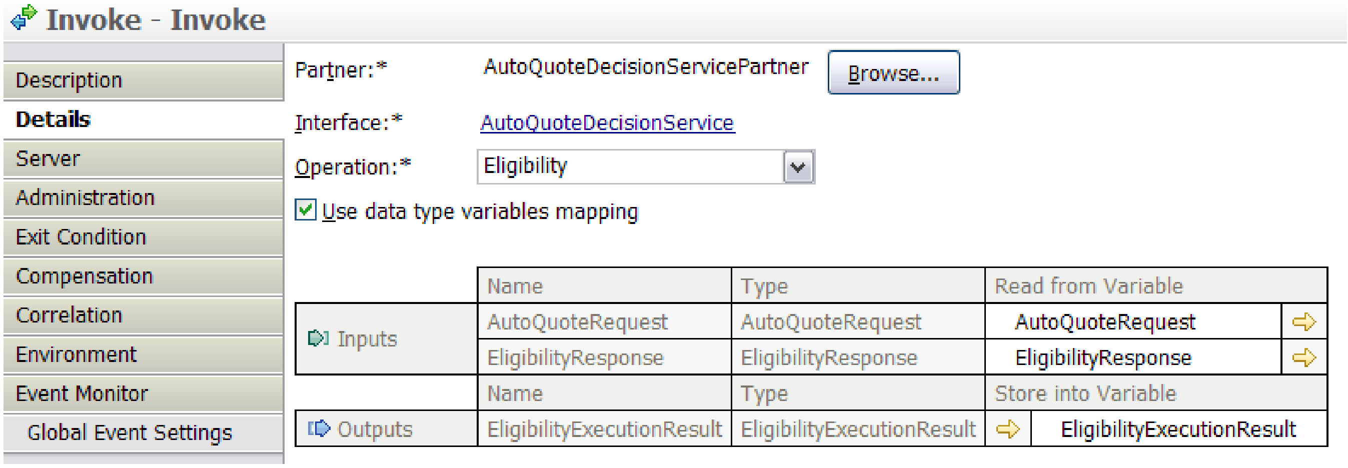
Select the Assign1 Node, and in the Properties screen, Details tab extract the **ValidationResponse** from the **DataValidationExecutionResult**.



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 undertaked for the Eligibility_AIS_Process
Create the **AutoQuoteDecisionServicePartner**.
Add the Invoke to the Process flow to invoke the **AutoQuoteDecisionService** creating the new variable **EligibilityExecutionResult**



Partner:* AutoQuoteDecisionServicePartner

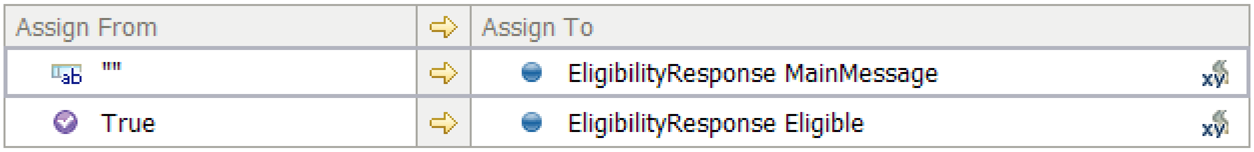
Interface:* AutoQuoteDecisionService

Operation:* Eligibility

☑ Use data type variables mapping

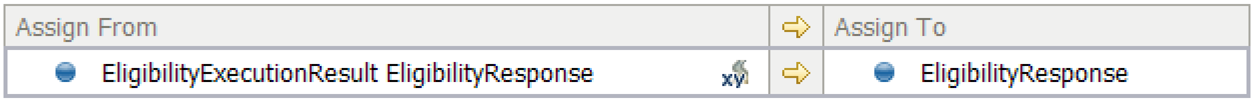
	Name	Type	Read from Variable	
Inputs	AutoQuoteRequest	AutoQuoteRequest	AutoQuoteRequest	⇨
	EligibilityResponse	EligibilityResponse	EligibilityResponse	⇨
	Name	Type	Store into Variable	
Outputs	EligibilityExecutionResult	EligibilityExecutionResult	⇨	EligibilityExecutionResult

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



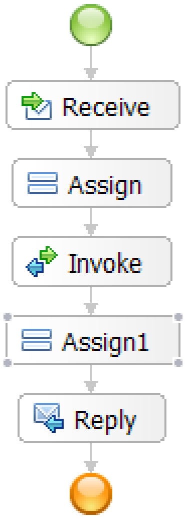
Assign From	⇨	Assign To
""	⇨	● EligibilityResponse MainMessage
True	⇨	● EligibilityResponse Eligible

Add the Assign1 to extract the EligibilityResponse from the EligibilityExecutionResult.



Assign From	⇨	Assign To
● EligibilityExecutionResult EligibilityResponse	⇨	● EligibilityResponse

The resulting process flow looks like this.

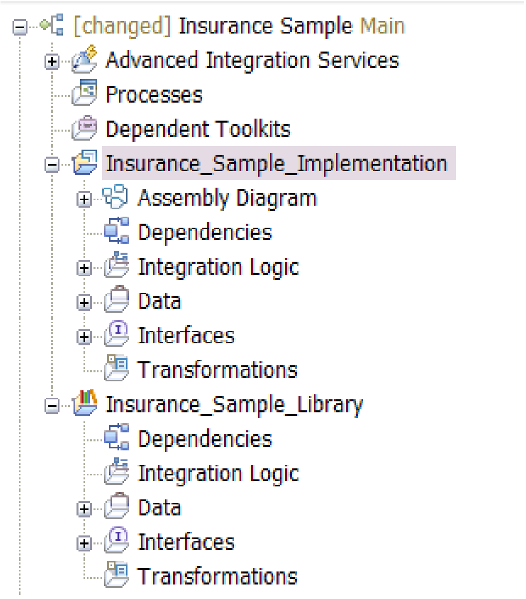


Clean and rebuild the projects and all errors should be removed.

Step 4. Defining the AIS Module Deployment Descriptor.

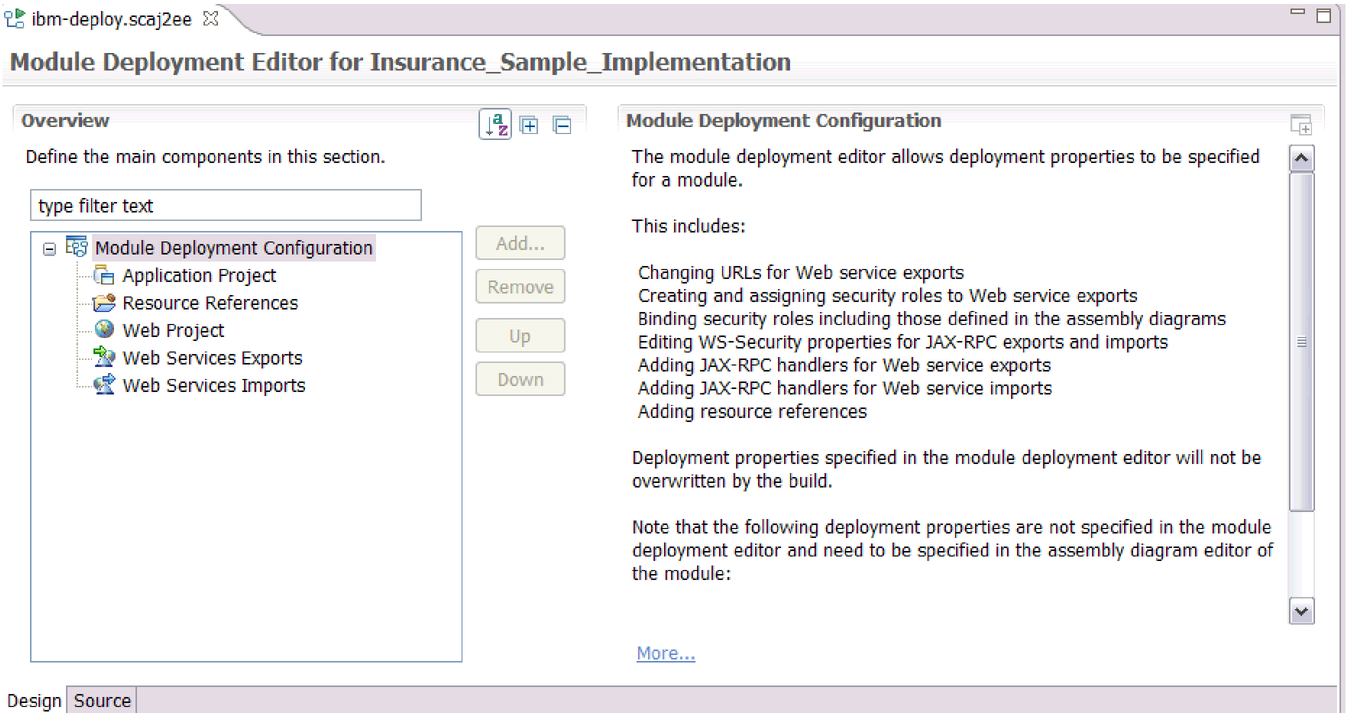
When working with AIS modules, the approach to defining the deployment descriptor is different. With previous tasks the deployment descriptor has been defined in the Web Application containing the module. With an AIS module, there is no containing module so the deployment descriptor has to be set up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


ibm-deploy.scaj2ee
Module Deployment Editor for Insurance_Sample_Implementation
Overview
Define the main components in this section.
type filter text
Module Deployment Configuration
Application Project
Resource References
Web Project
Web Services Exports
Web Services Imports
Add...
Remove
Up
Down
Module Deployment Configuration

The module deployment editor allows deployment properties to be specified for a module.

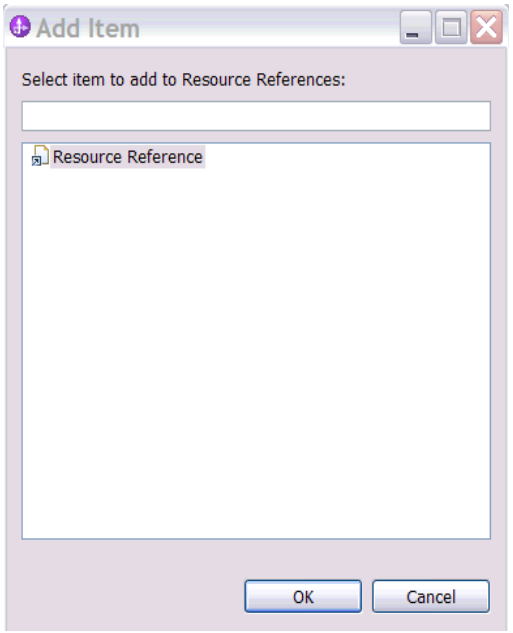
This includes:

Changing URLs for Web service exports
Creating and assigning security roles to Web service exports
Binding security roles including those defined in the assembly diagrams
Editing WS-Security properties for JAX-RPC exports and imports
Adding JAX-RPC handlers for Web service exports
Adding JAX-RPC handlers for Web service imports
Adding resource references

Deployment properties specified in the module deployment editor will not be overwritten by the build.

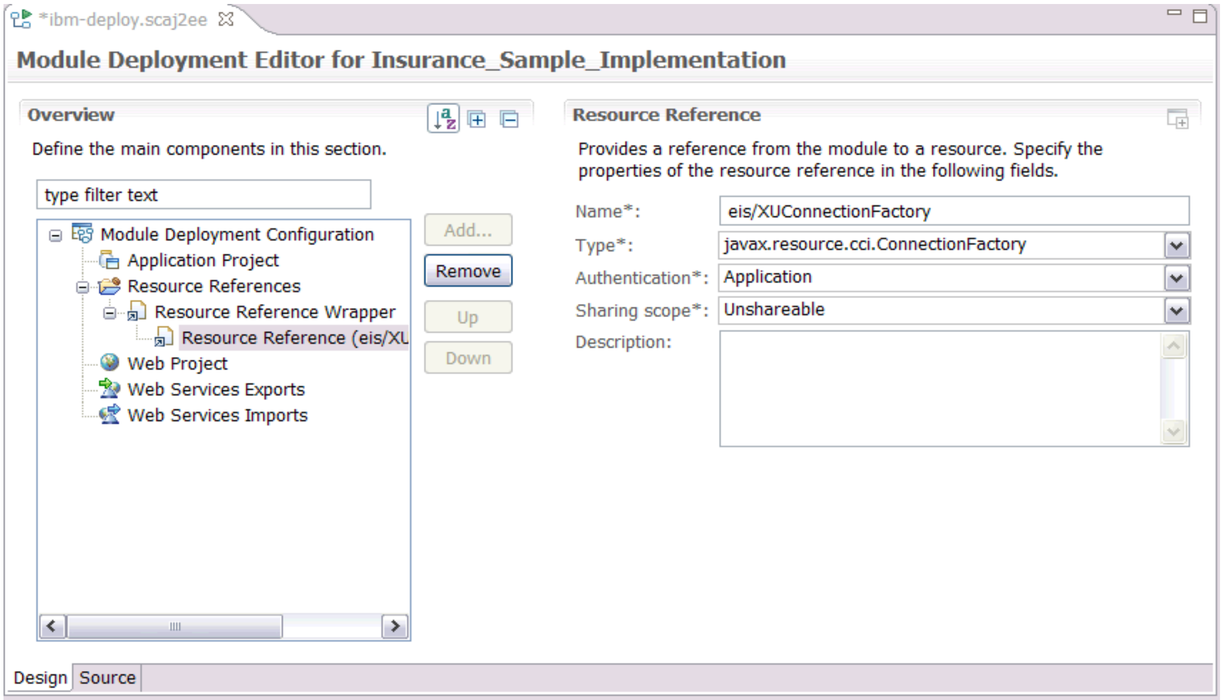
Note that the following deployment properties are not specified in the module deployment editor and need to be specified in the assembly diagram editor of the module:

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.

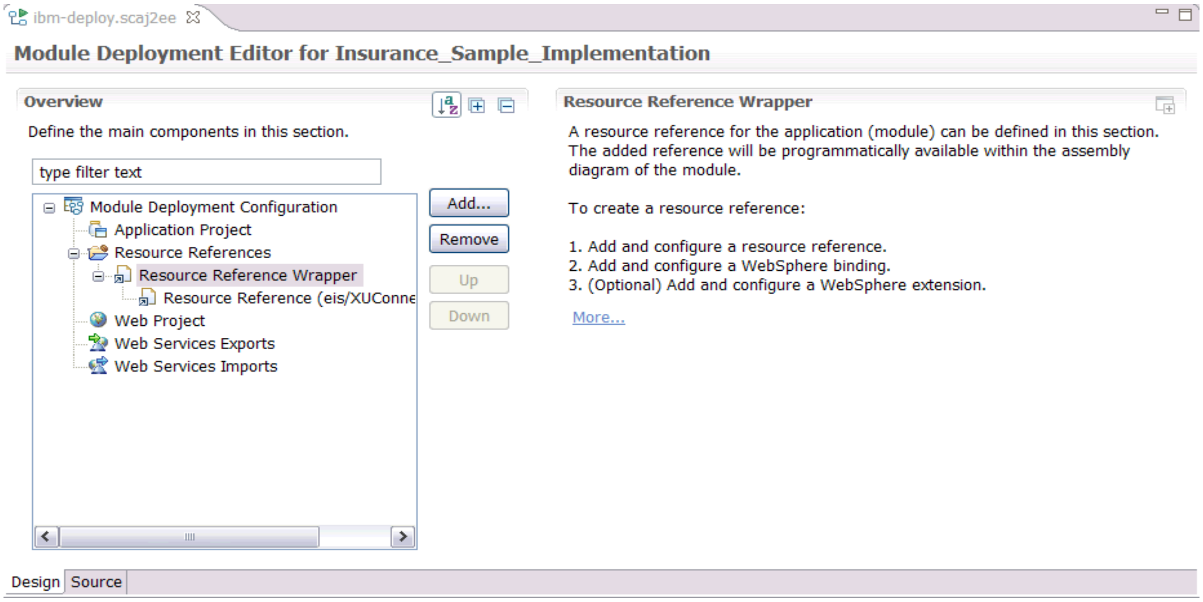


Define the main components in this section.

Provides a reference from the module to a resource. Specify the properties of the resource reference in the following fields.

Name*: eis/XUConnectionFactory
Type*: javax.resource.cci.ConnectionFactory
Authentication*: Application
Sharing scope*: Unshareable
Description:

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


ibm-deploy.scaj2ee
Module Deployment Editor for Insurance_Sample_Implementation
Overview
type filter text
Module Deployment Configuration
Application Project
Resource References
Resource Reference Wrapper
Resource Reference (eis/XUConne
Web Project
Web Services Exports
Web Services Imports
Add...
Remove
Up
Down
Resource Reference Wrapper
More...
Design
Source

Define the main components in this section.

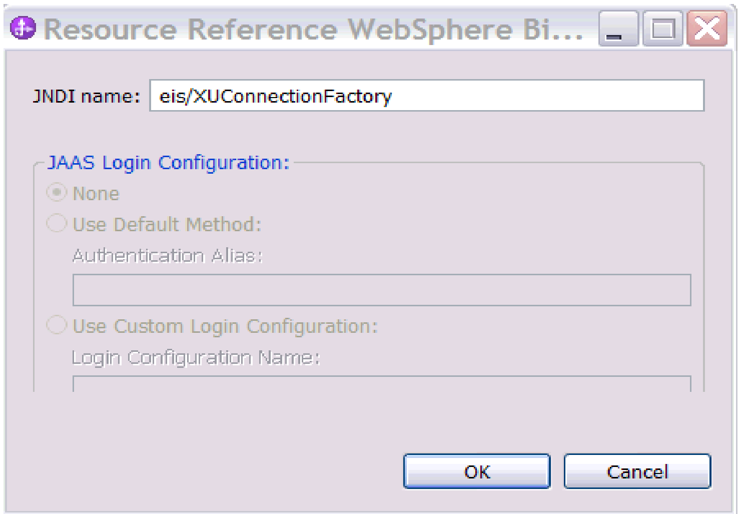
A resource reference for the application (module) can be defined in this section. The added reference will be programmatically available within the assembly diagram of the module.

To create a resource reference:

1. Add and configure a resource reference.
2. Add and configure a WebSphere binding.
3. (Optional) Add and configure a WebSphere extension.

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

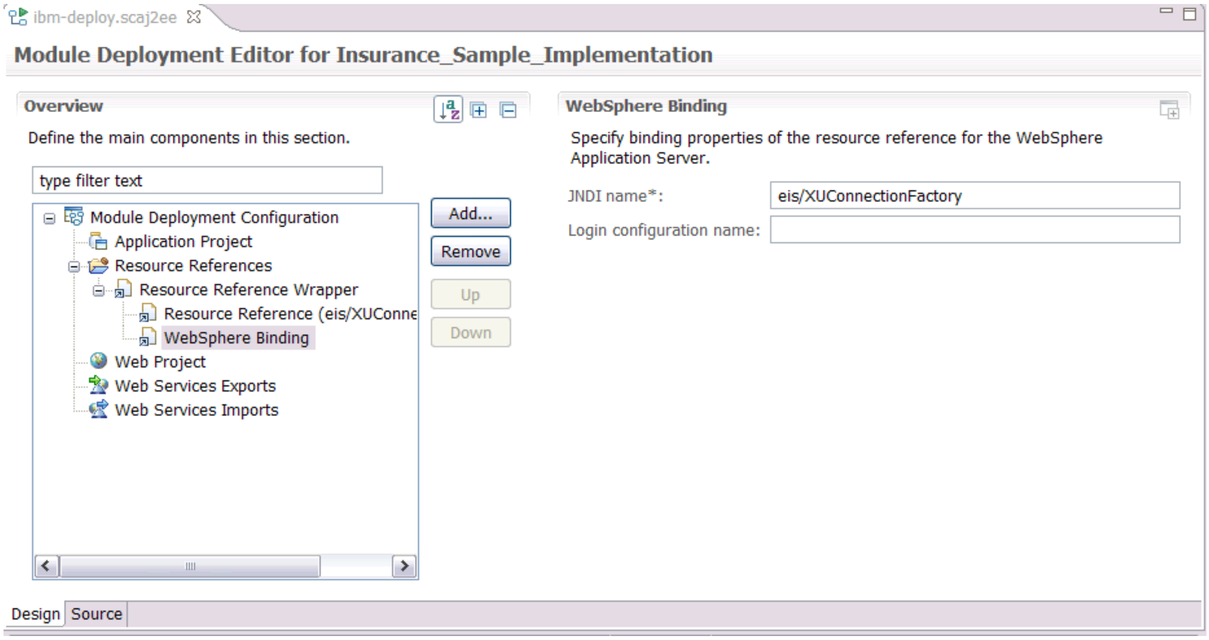
In the Resource Reference WebSphere Binding popup enter the JNDI name as shown below.


Resource Reference WebSphere Bi...
JAAS Login Configuration:
○ Use Default Method:
Authentication Alias:
○ Use Custom Login Configuration:
Login Configuration Name:
OK
Cancel

JNDI name: eis/XUConnectionFactory

● None

Click OK.



Module Deployment Editor for Insurance_Sample_Implementation

WebSphere Binding

Specify binding properties of the resource reference for the WebSphere Application Server.

JNDI name*: eis/XUConnectionFactory

Login configuration name:

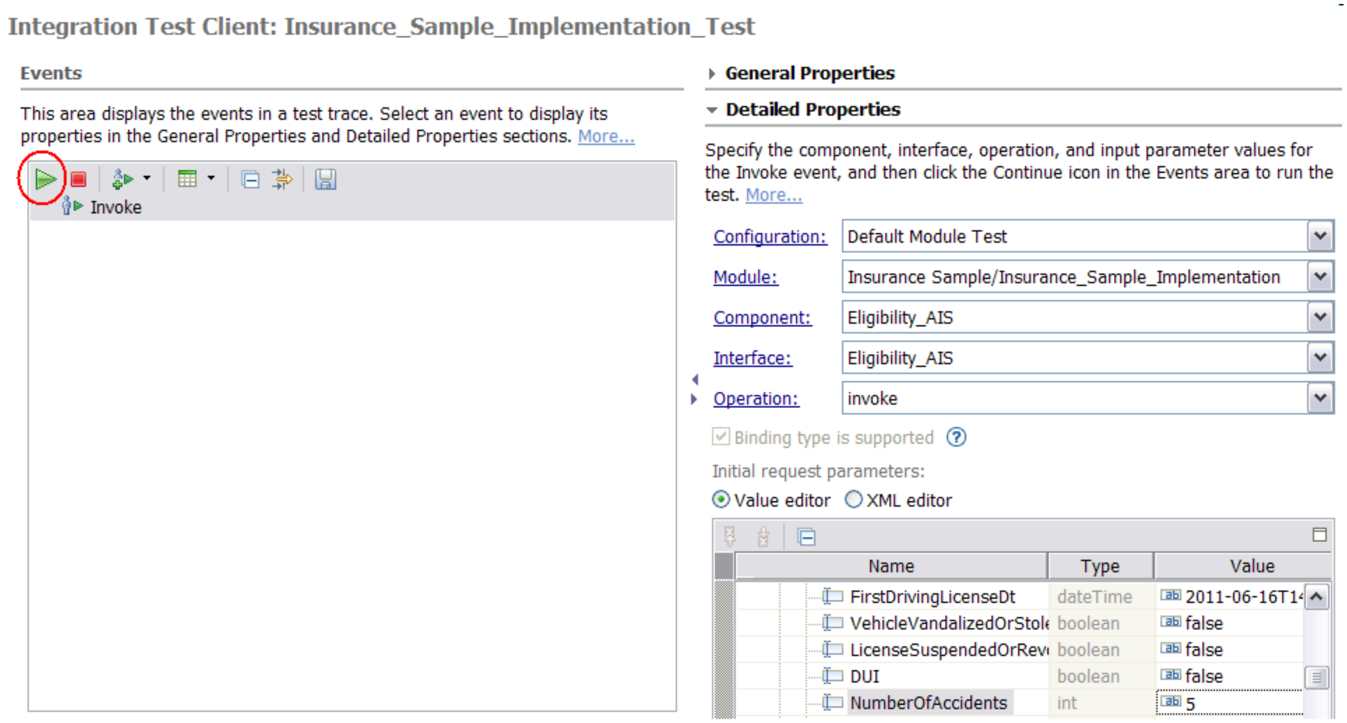
You have now configured the SCA module to refer to the eXection Unit deployed on the same server as the module itself.

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...

Invoke

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

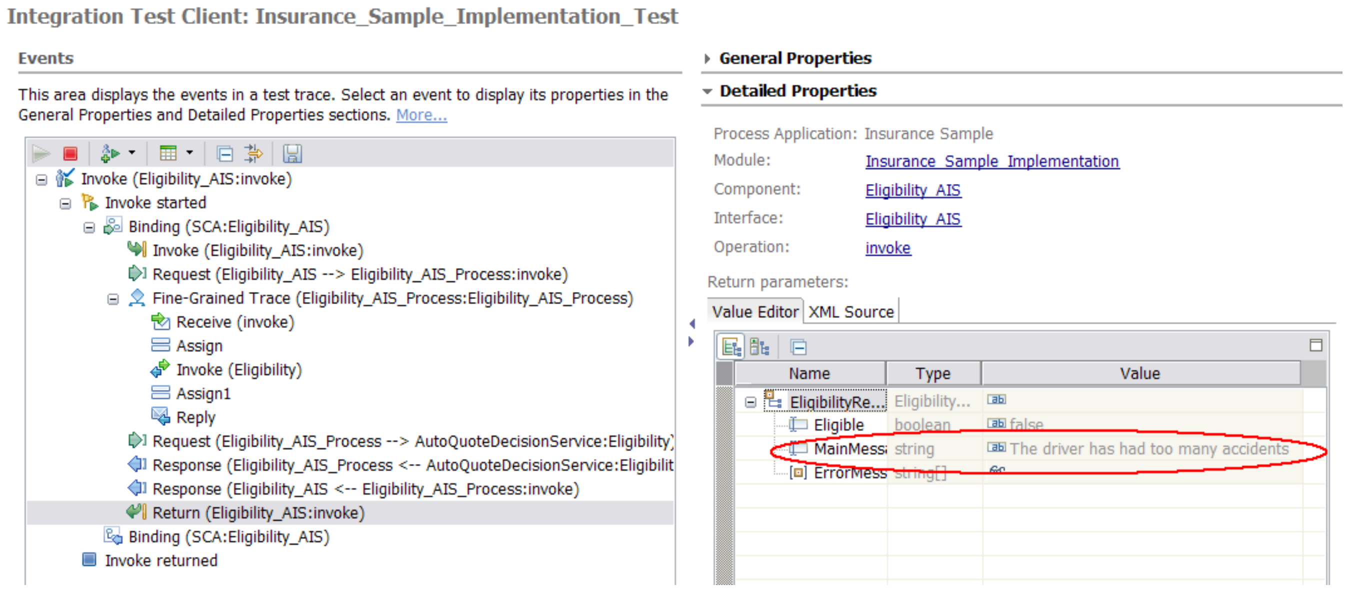
☑ Binding type is supported

Initial request parameters:
● Value editor ○ XML editor

Name	Type	Value
FirstDrivingLicenseDt	dateTime	2011-06-16T1
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 **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...

- Invoke (Eligibility_AIS:invoke)
  - Invoke started
    - Binding (SCA:Eligibility_AIS)
      - Invoke (Eligibility_AIS:invoke)
      - Request (Eligibility_AIS --> Eligibility_AIS_Process:invoke)
      - Fine-Grained Trace (Eligibility_AIS_Process:Eligibility_AIS_Process)
        - Receive (invoke)
        - Assign
        - Invoke (Eligibility)
        - Assign1
        - Reply
      - Request (Eligibility_AIS_Process --> AutoQuoteDecisionService:Eligibility
      - Response (Eligibility_AIS_Process <-- AutoQuoteDecisionService:Eligibilit
      - Response (Eligibility_AIS <-- Eligibility_AIS_Process:invoke)
      - Return (Eligibility_AIS:invoke)
    - Binding (SCA:Eligibility_AIS)
  - Invoke returned

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 SupportPacLA71v1.7\BPMTutorial\task7\Insurance_Sample – LA71TT7.twx

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