

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

Getting started with IBM Business Process Manager

Task 5 – Integration Designer assembles an SCA Decision Component from a Managed Decision RuleApp with SupportPac LA71

© Copyright IBM Corp. 1987, 2013

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

Copyright

Copyright notice

© Copyright IBM Corp. 1987, 2013

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSAADP 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 are going to use the LA71 wizard to create an SCA Managed Decision component from the RuleApp archive you exported earlier. You will then test the decision service using the Integrated Test Component. Two implementations are described: The POJO Implementation has the best performance when the Rule Execution Server deployed to the Process Server Application server. The EJB implementation supports the same interfaces but allows rules to be deployed on a separate Application server.

Readers who want to skip the exercise should import the solution into IID

[SupportPac LA71 Path]\BPMTutorial\task5\BPELProcessDecisionModule.zip and continue at Step 4 to test the POJO SCA Decision Component in the Process Server. You can also test the EJB SCA Decision Component as described in Step4.

Step 1. Create the BPEL Process Module.

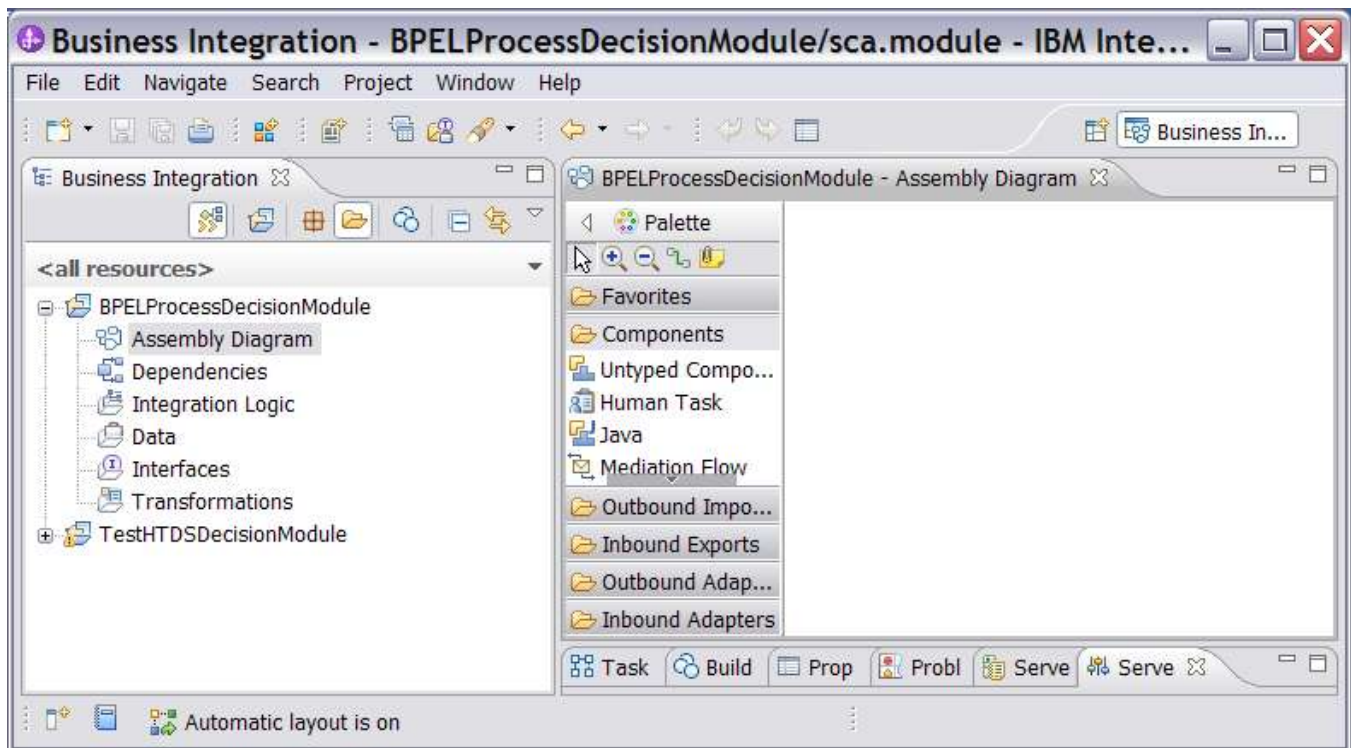
In this step we will create a module that contains the SCA Decision Component and will eventually contain the BPEL Process that uses it.

Open Integration Designer.

Click **File** → **New** → **Module** from the menu.

Give the module the following name: **BPELProcessDecisionModule**.

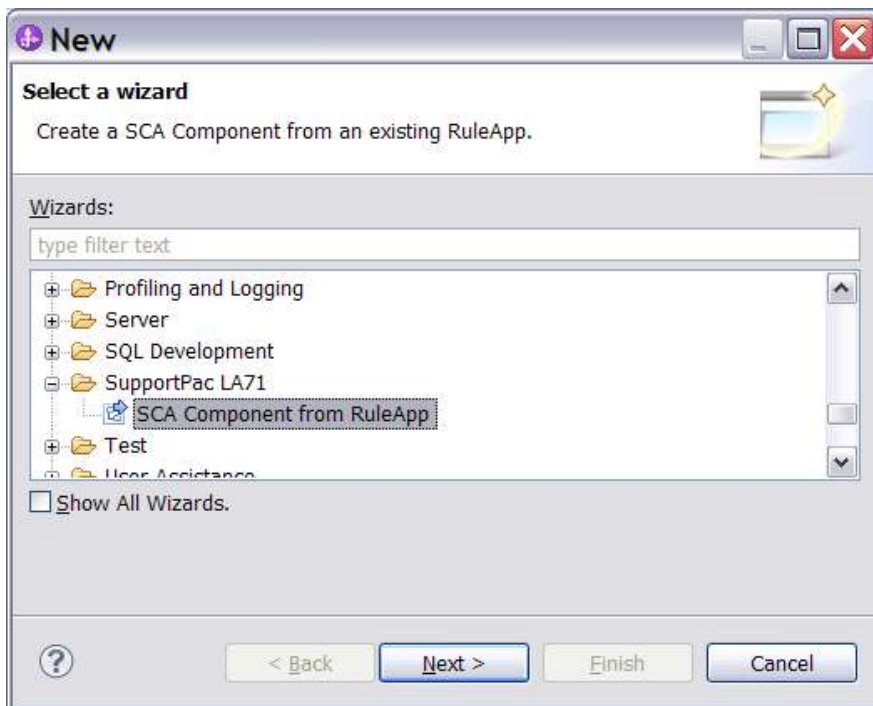
Click **Finish**.



Step 2. Use LA71 wizard to create the Decision Component using POJO implementation.

Select **File > New > Other...**

In the wizard that opens Expand **SupportPac LA71** and select **SCA Component from RuleApp**.



Click **Next**.

Alternatively you can right-click the module and select **SupportPac LA71 > Create SCA Component from RuleApp**.

In the panel that appears:

In the Project Field select **BPELProcessDecisionModule**

Click the **Archive** button to browse to the folder where you saved the RuleApp archive or the completed **[SupportPac LA71 Path]\BPMTutorial\task2** directory and select

InsuranceSampleRuleApp.jar, and then press **Open**.

Ensure that the checkboxes next to the **DataValidation** and **Eligibility** rulesets are checked and then click **Next >**.

Business Decision Wizard

RuleApp Selection

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

Click Project to select a Module Project

BPELProcessDecisionModule Project...

Click Project or Archive to select RuleApp

ram Files\IBM\ODM801\SupportPacLA71v2.1\BPMTutorial\InsuranceSampleRuleApp.jar Archive...

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

☒ DataValidation
 ☒ Eligibility

Input Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationResponse	ValidationResponse

Output Parameters:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationResponse	ValidationResponse

? < Back Next > Finish Cancel

In this page of the wizard, there are some warnings in the bottom Task List due to unresolved types in the ruleset signatures. In some cases you need to provide java libraries as well as XML schemas used in the Rule Projects.

You will need to identify the schemas that define the business objects and BOM from the RuleWorkspace.

Business Decision Wizard

Object Model Mapping
Add Java Libraries and XML Schemas for Data Mapping.

Java libraries to be included in the project:

Jar Library	Location

Add External JARs
Add JARs...
Remove

XML Schemas to be included in the project:

XML Schema	Location

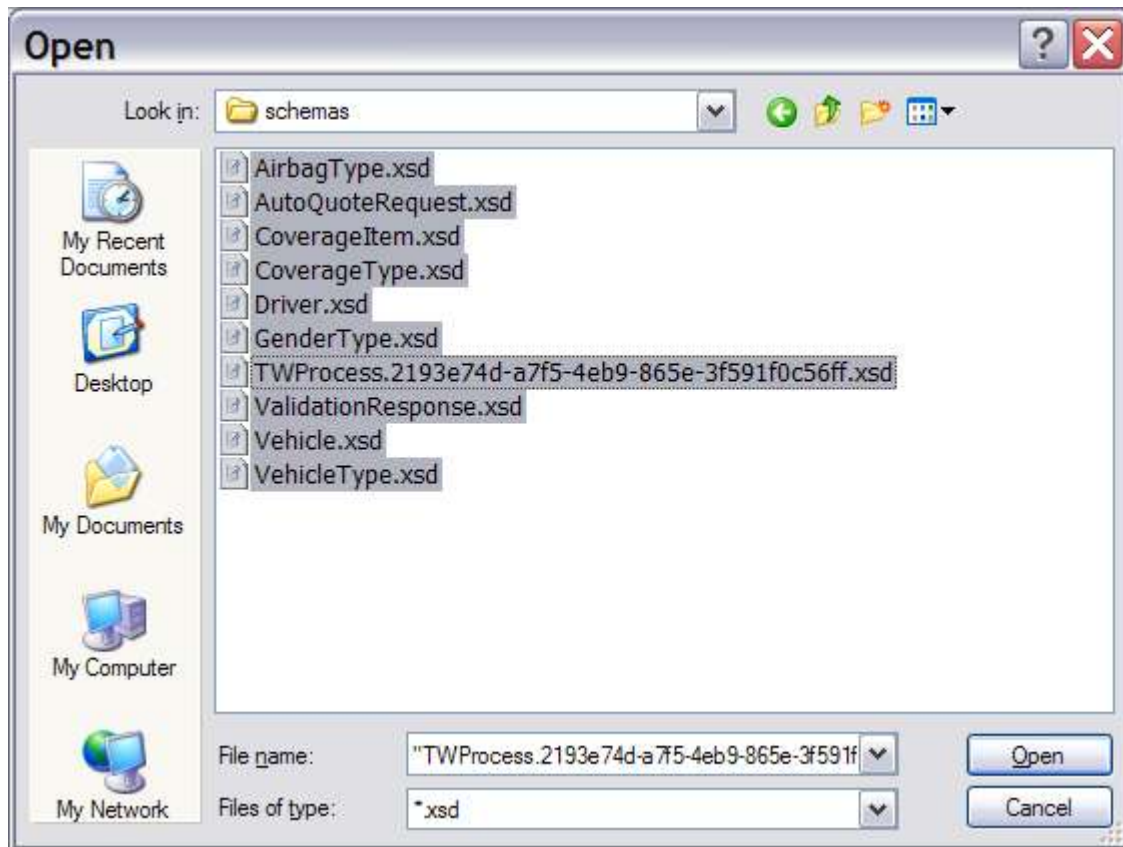
Add External XSDs
Add XSDs...
Remove

Tasks:

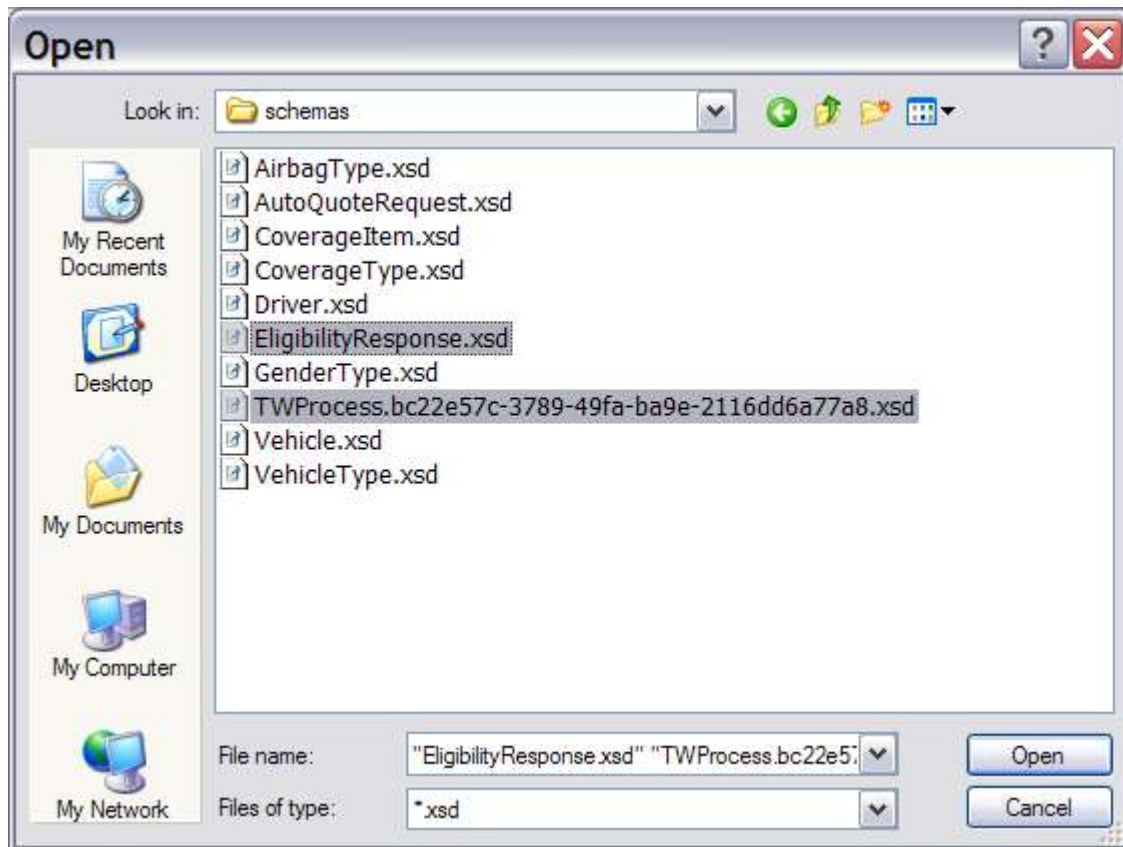
Description	Ruleset
⚠ {http://INSDemo}AutoQuoteRequest cannot be resolved	DataValidati...
⚠ {http://INSDemo}ValidationResponse cannot be resolved	DataValidati...
⚠ {http://INSDemo}EligibilityResponse cannot be resolved	Eligibility

? < Back Next > Finish Cancel

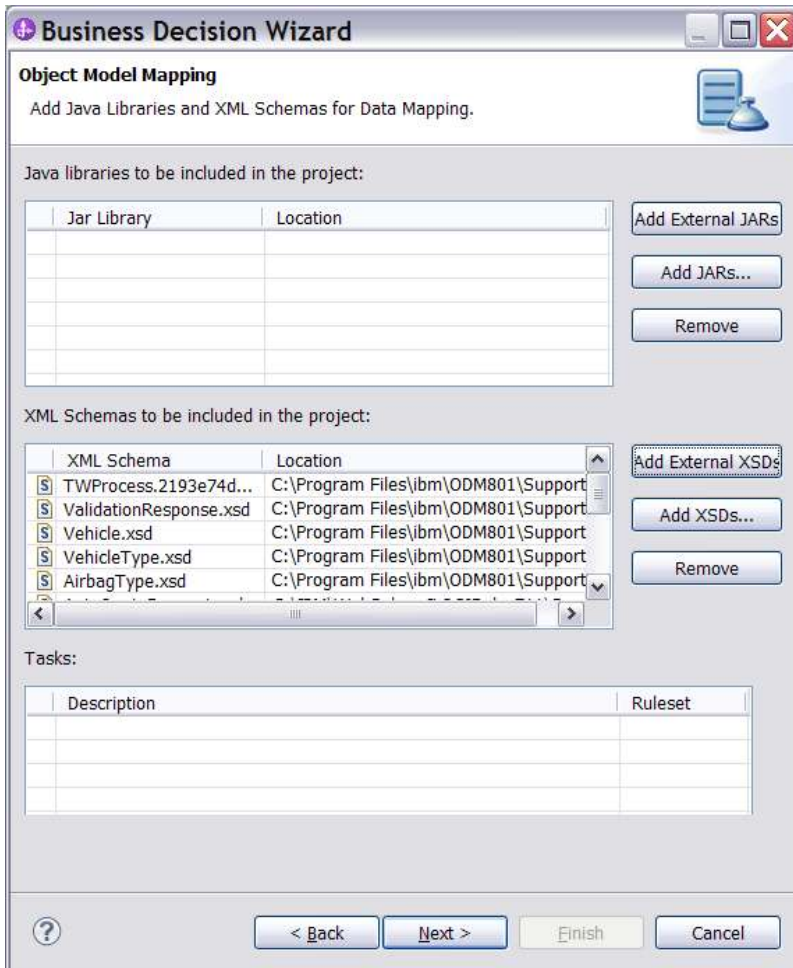
Click the **Add External XSDs** button and browse first to the the **[SupportPac LA71 Path]** **\BPMTutorial\RuleWorkspace\DataValidation\schemas** directory.. Select all schemas in this directory as shown.



Click **Open** and then click the **Add External XSDs** button again. Browse to the **[SupportPac LA71 Path]\BPMTutorial\RuleWorkspace\Eligibility\schemas** directory. Select only the root schema and those schemas unique to the Eligibility decision.



Click **Open** and check that the warnings have all been resolved and that the **Next** button has appeared.



Click **Next** and In the next panel :
Set the **Package** field to **bpelprocess**.

Set the **Name** field to **ManagedDecision**.

Set the **Implementation Type** to **POJO Invocation**. It is recommended to use the POJO invocation approach for this component since we have the Rule Execution Server installed on Business Process Manager and can take advantage of the proximity to make a direct Java call rather than going through the EJB Container.

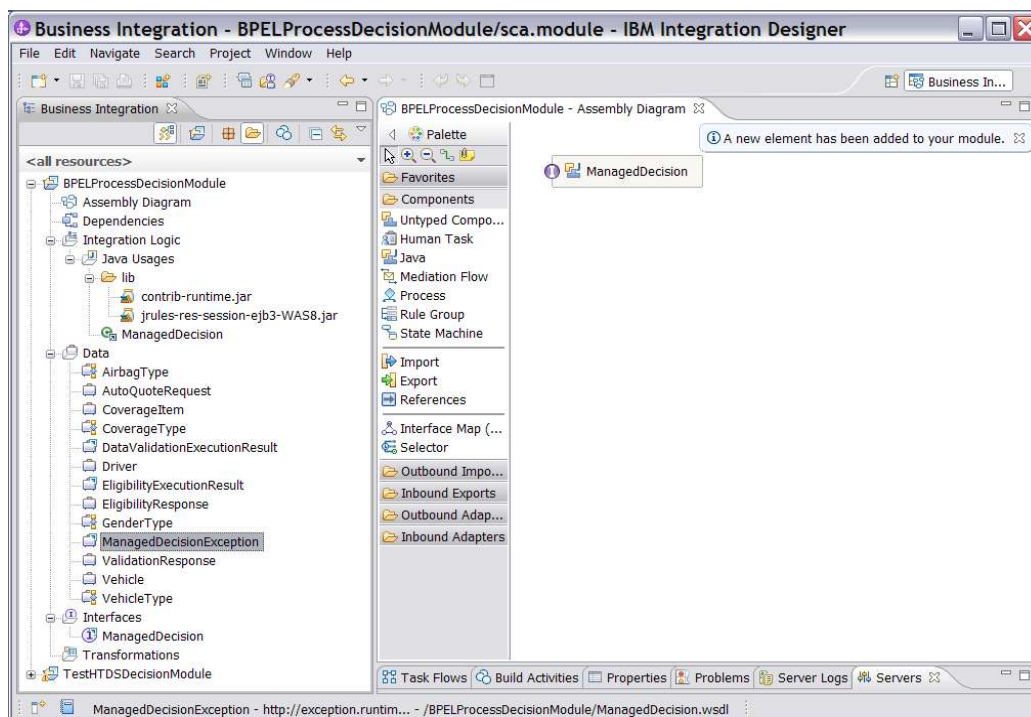
Other fields should be set automatically.

The wizard will generate Java code to invoke the latest version of the identified ruleset using the J2EE POJO interface. This invokes the Rule Execution Server eXecution Unit (XU) Resource Adapter providing scalable rule execution within the Application server.



Click **Finish**.

The workspace should now show the Module Assembly Diagram with the new ManagedDecision interface available.



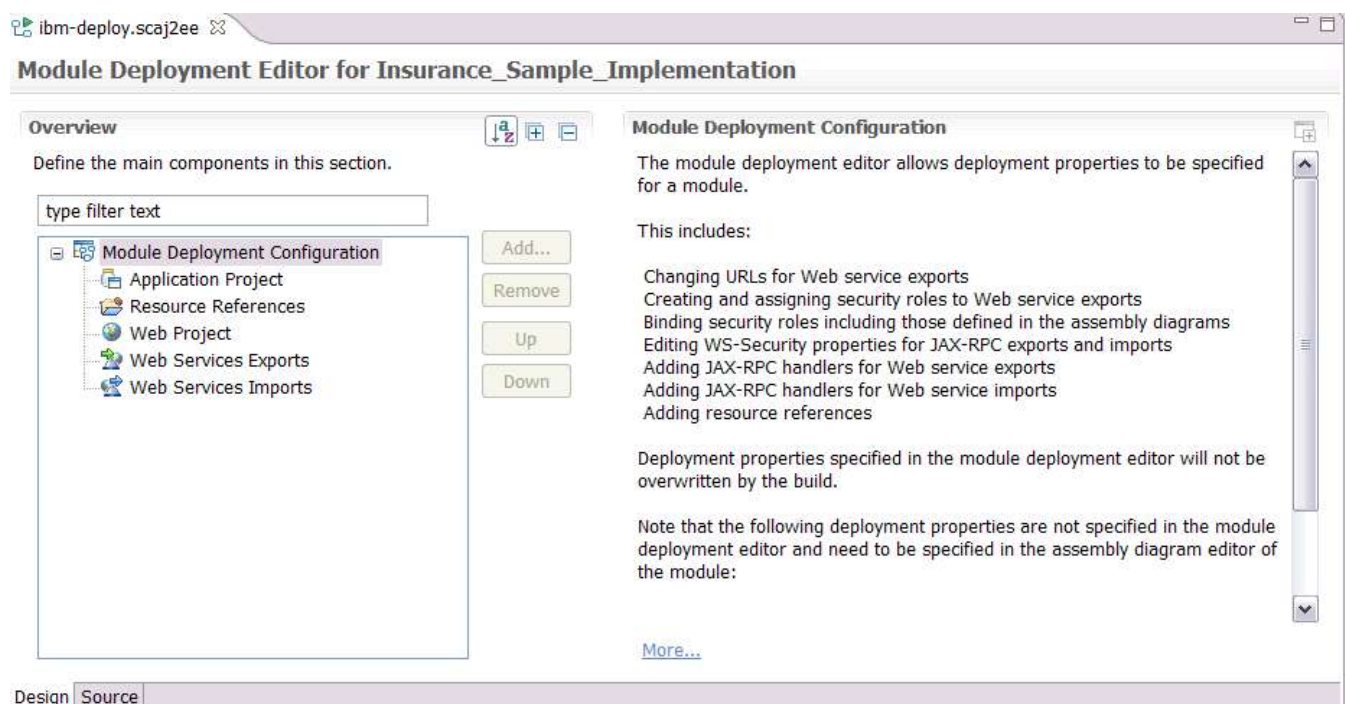
Step 3. Defining the Module Deployment Descriptor.

When working with the SCA module for EJB session, you need to set up the JNDI name and its reference for the rule execution server.

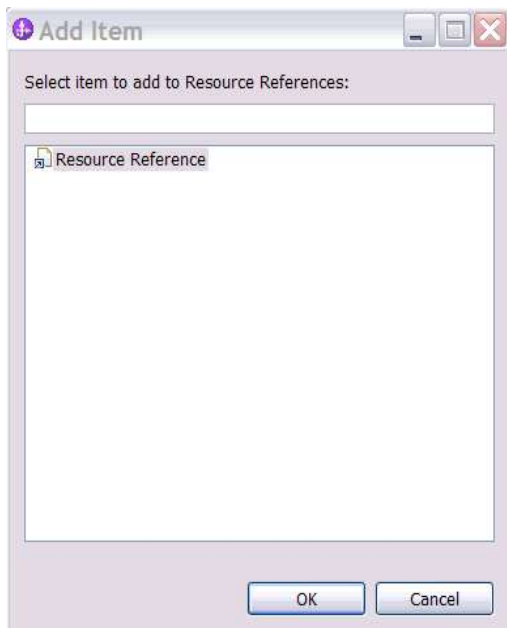
To setup the deployment descriptor

Select the **BPELProcessDecisionModule** module, right click and select **Open Deployment Editor**

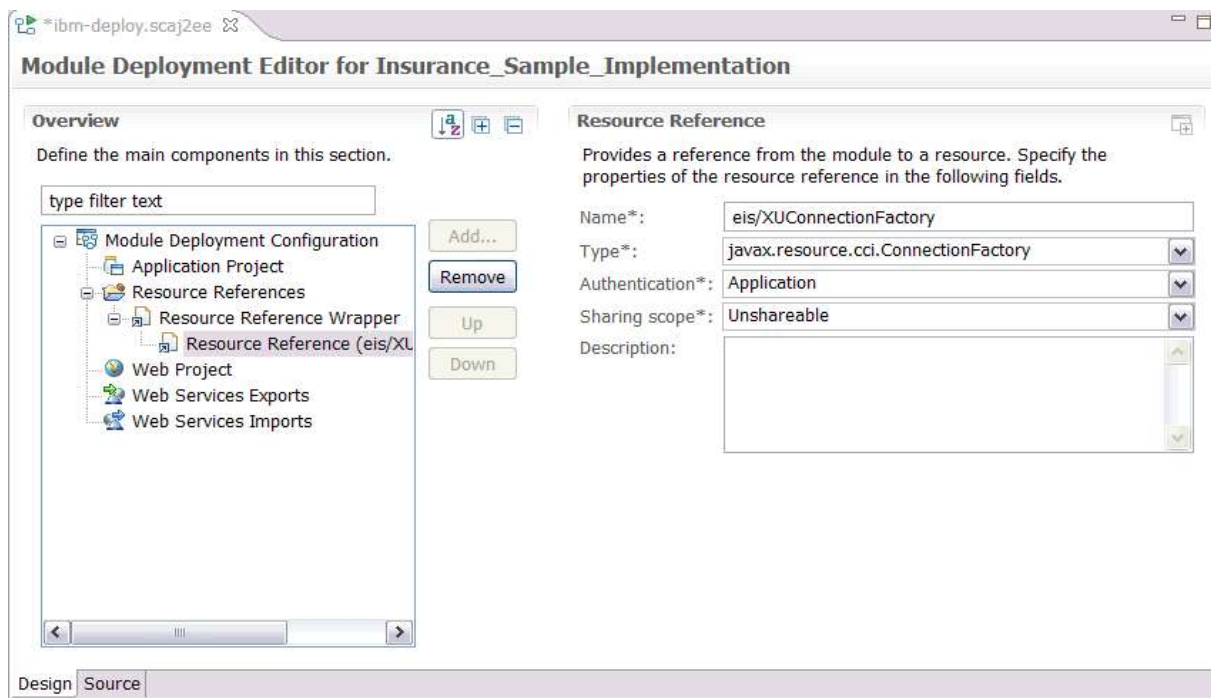
This will open up a deployment editor for the SCA Module as shown below.



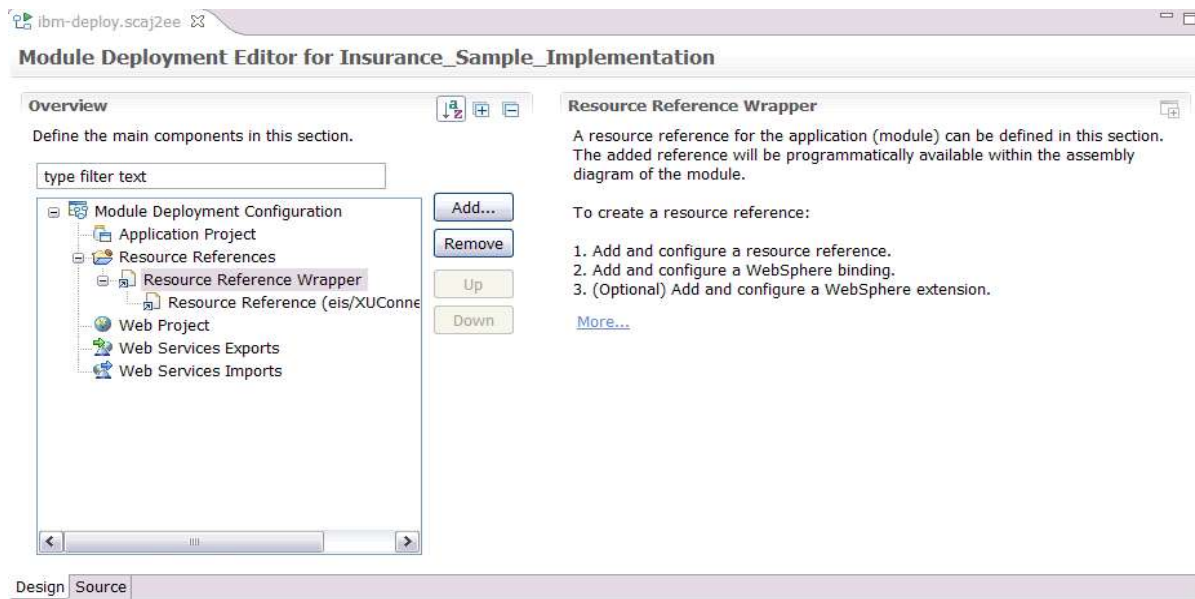
Select **Resource References** and click **Add...**



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.

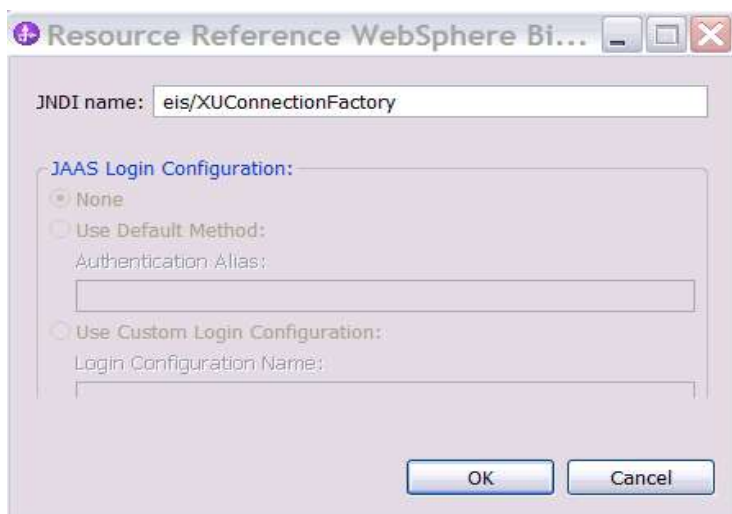


Save your work and select the **Resource Reference Wrapper** as shown below.

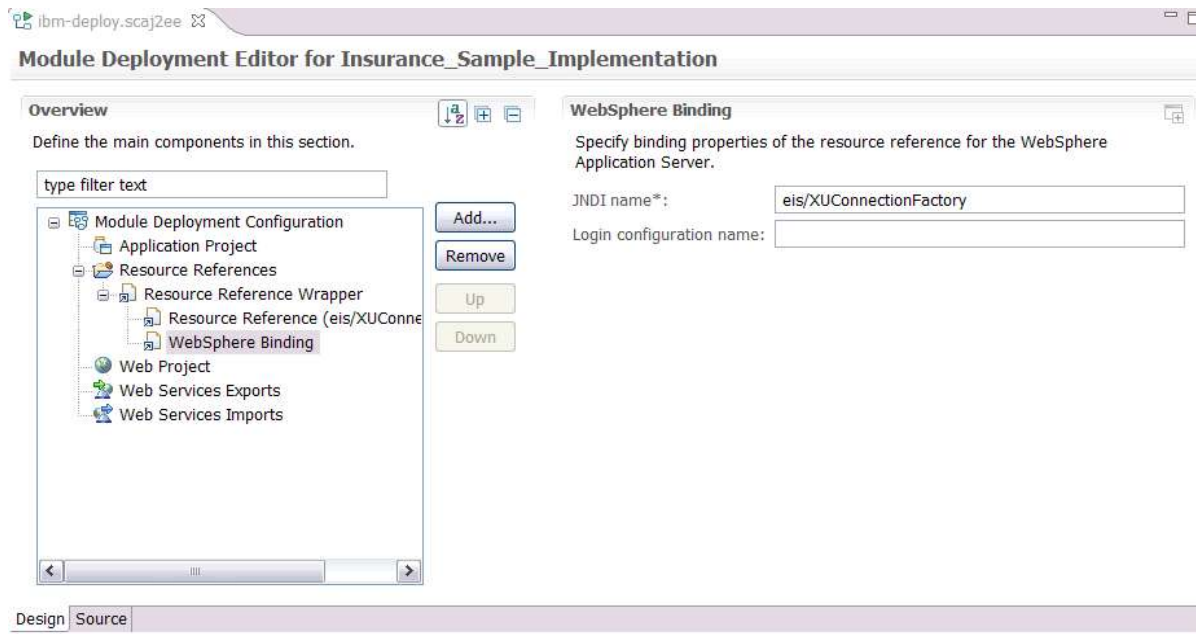


To create the WebSphere binding right click the Resource Reference Wrapper and Select **Add > WebSphere Binding**.

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 4. Test the POJO Decision Component

With the **ManagedDecision** SCA Component created, you can now test the calls to the RuleApp. Associate your SCA component to a process application and open the Assembly Diagram if it is not open already.

Right-click on the **ManagedDecision** and select **Test Component**.

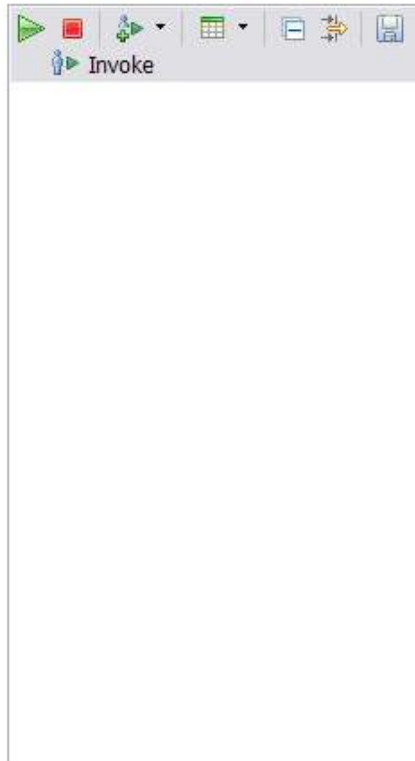
For the **Operation:**, select **Eligibility**.

In the **Value editor**, navigate to the **NumberOfAccidents** property and set the value to **5** which should cause the rules to indicate that the quote is not eligible.

Integration Test Client: BPELProcessDecisionModule_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...](#)



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: BPELProcessDecisionModule

Component: ManagedDecision

Interface: ManagedDecision

Operation: Eligibility

Initial request parameters:

☒ Value editor ☐ XML editor

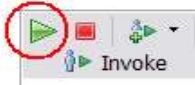
Name	Type	Value
VehicleVandalizedOrS	boolean	false
LicenseSuspendedOrI	boolean	false
DUI	boolean	false
NumberOfAccidents	int	5
NumberOrI traffic tcke	int	0
Vehicle	Vehicle	
VehicleIdentificationN	string	

To edit values, start typing or press F2.

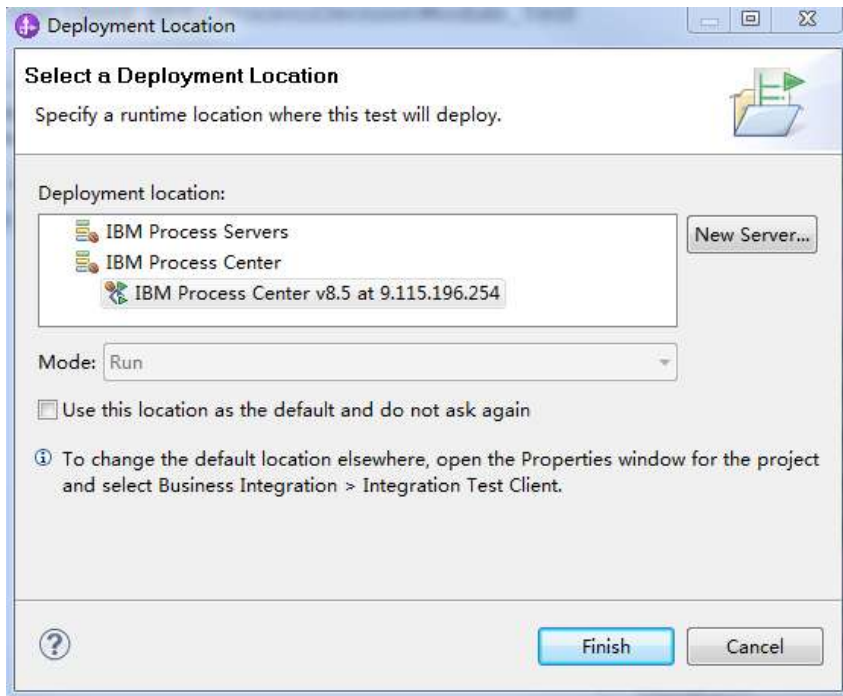
Select the continue icon.

Events

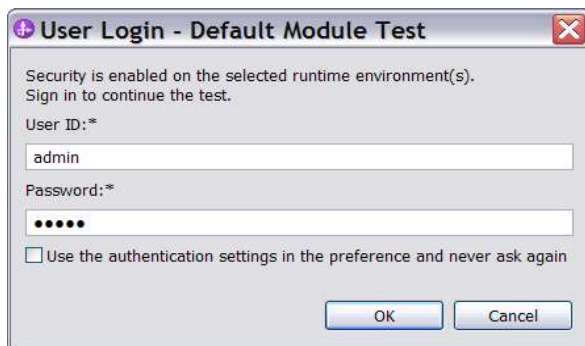
This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. [More...](#)



In the **Select a Deployment Location** screen select your **IBM Process Center** and click **Finish**.



In the User Login screen, enter the administrator name and password for the environment
For **User ID**: type **admin**
For **Password**: type **admin**



The module is published to the server, and the request data sent to the module. The module then invokes the Eligibility Decision Service and gets the response back which is then made available as the return business object. The rules have executed and indicated that the quote is not eligible because the number of accidents was too high.

Integration Test Client: BPELProcessDecisionModule_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

Module: [BPELProcessDecisionModule](#)

Component: [ManagedDecision](#)

Interface: [ManagedDecision](#)

Operation: [Eligibility](#)

Return parameters:

Value Editor XML Source

Name	Type	Value
VehicleType	VehicleType<string>	abi
AirbagType	AirbagType<string>	abi
AntiLockBrakes	boolean	abi false
DayTimeRunnir	boolean	abi false
Make	string	abi
Model	string	abi
ModelYear	int	abi 0
Value	double	abi 0.0
Coverage	CoverageItem[]	abi
EligibilityResponse *	EligibilityResponse	abi
Eligible	boolean	abi false
MainMessage	string	abi The driver has had too many accidents
ErrorMessages	string[]	abi

You have now successfully created and tested an SCA Managed Decision Component using a POJO session interface.

Step 5. Use LA71 wizard to create the Decision Component using EJB interface.

In this optional step you will repeat the exercise undertaken in steps 2 and 3 but will generate the module using an EJB implementation to connect to the Rule Execution Server. Instead of interacting with the XU Resource adapter, the LA71 code generated will invoke a stateless session bean that has been deployed on the server.

Note that the stateless session bean must be deployed **manually**. Please ensure that the RES EJB Session Jar has been installed into WAS according to the instructions in **BPM Tutorial Overview**, otherwise the module using EJB implementation won't work properly.

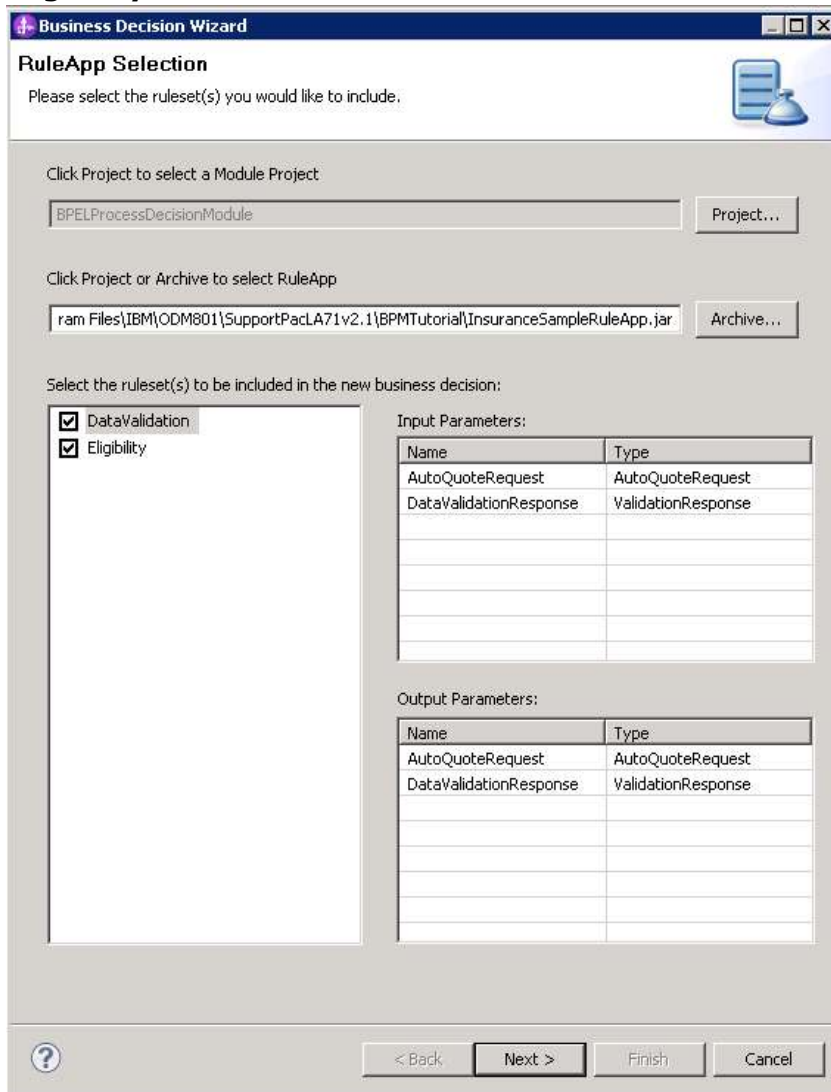
Right click the **BPELProcessDecisionModule** Module.

Select **SupportPac LA71 > Create SCA Component from RuleApp**

In the panel that appears:

In the Project Field ensure **BPELProcessDecisionModule** is selected

Click the **Archive...** button to 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.



The dialog box is titled "Business Decision Wizard" and "RuleApp Selection". It contains the following elements:

- A text box for "Click Project to select a Module Project" with the value "BPELProcessDecisionModule" and a "Project..." button.
- A text box for "Click Project or Archive to select RuleApp" with the value "ram Files\IBM\ODM801\SupportPacLA71v2.1\BPMTutorial\InsuranceSampleRuleApp.jar" and an "Archive..." button.
- A section "Select the ruleset(s) to be included in the new business decision:" with a list box containing "DataValidation" and "Eligibility", both of which are checked.
- A section "Input Parameters:" with a table:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationResponse	ValidationResponse

- A section "Output Parameters:" with a table:

Name	Type
AutoQuoteRequest	AutoQuoteRequest
DataValidationResponse	ValidationResponse

At the bottom, there is a help icon (?) and four buttons: "< Back", "Next >", "Finish", and "Cancel".

Click **Next**.

The schemas required should already be resolved from the import for the POJO implementation. Check that there are no warnings and that the **Next** button has appeared.

Business Decision Wizard

Object Model Mapping

Add Java Libraries and XML Schemas for Data Mapping.

Java libraries to be included in the project:

Jar Library	Location

Add External JARs
Add JARs...
Remove

XML Schemas to be included in the project:

XML Schema	Location

Add External XSDs
Add XSDs...
Remove

Tasks:

Description	Ruleset

? < Back Next > Finish Cancel

Click **Next** and in the **Business Decisions Settings** panel :

Set the **Package** field to **bpelprocess**.

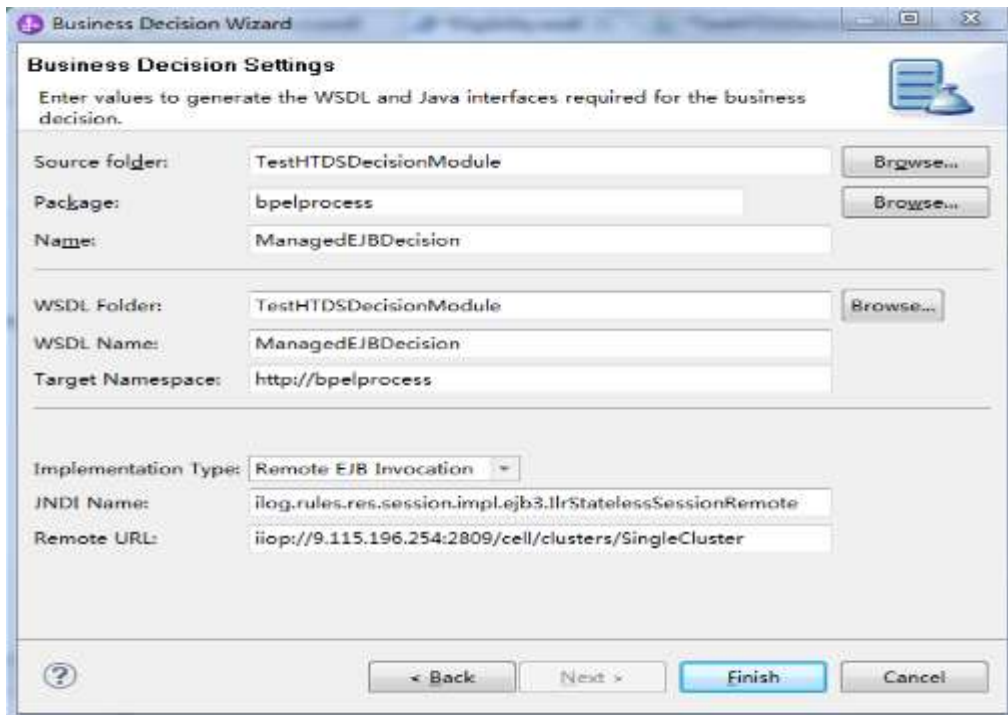
Set the **Name** field to **ManagedEJBDecision**.

Set the **Implementation Type** to **Remote EJB Invocation**.

Leave the **JNDI Name** field as it is to connect to the default JRules Session bean.

Set the **Remote Url** to the IIOP Url of the server hosting the JRules Session bean – **iiop://<host_name>:2809/cell/clusters/SingleCluster** for cluster environment.

The wizard will generate Java code to invoke the latest version of the identified ruleset using the J2EE Remote EJB interface. This invokes the Rule Execution Server stateless session EJB providing scalable rule execution on a remote or local Application server.



The image shows a 'Business Decision Wizard' dialog box with the title 'Business Decision Settings'. It contains several input fields and buttons. The fields are: 'Source folder:' with value 'TestHTDSDDecisionModule', 'Package:' with value 'bpelprocess', 'Name:' with value 'ManagedEJBDecision', 'WSDL Folder:' with value 'TestHTDSDDecisionModule', 'WSDL Name:' with value 'ManagedEJBDecision', 'Target Namespace:' with value 'http://bpelprocess', 'Implementation Type:' with a dropdown menu showing 'Remote EJB Invocation', 'JNDI Name:' with value 'ilog.rules.res.session.impl.ejb3.IlrStatelessSessionRemote', and 'Remote URL:' with value 'iiop://9.115.196.254:2809/cell/clusters/SingleCluster'. There are 'Browse...' buttons next to the 'Source folder:', 'Package:', and 'WSDL Folder:' fields. At the bottom, there are buttons for '< Back', 'Next >', 'Finish' (highlighted in blue), and 'Cancel'. A help icon (?) is also present in the bottom left corner.

Business Decision Settings

Enter values to generate the WSDL and Java interfaces required for the business decision.

Source folder: TestHTDSDDecisionModule Browse...

Package: bpelprocess Browse...

Name: ManagedEJBDecision

WSDL Folder: TestHTDSDDecisionModule Browse...

WSDL Name: ManagedEJBDecision

Target Namespace: http://bpelprocess

Implementation Type: Remote EJB Invocation

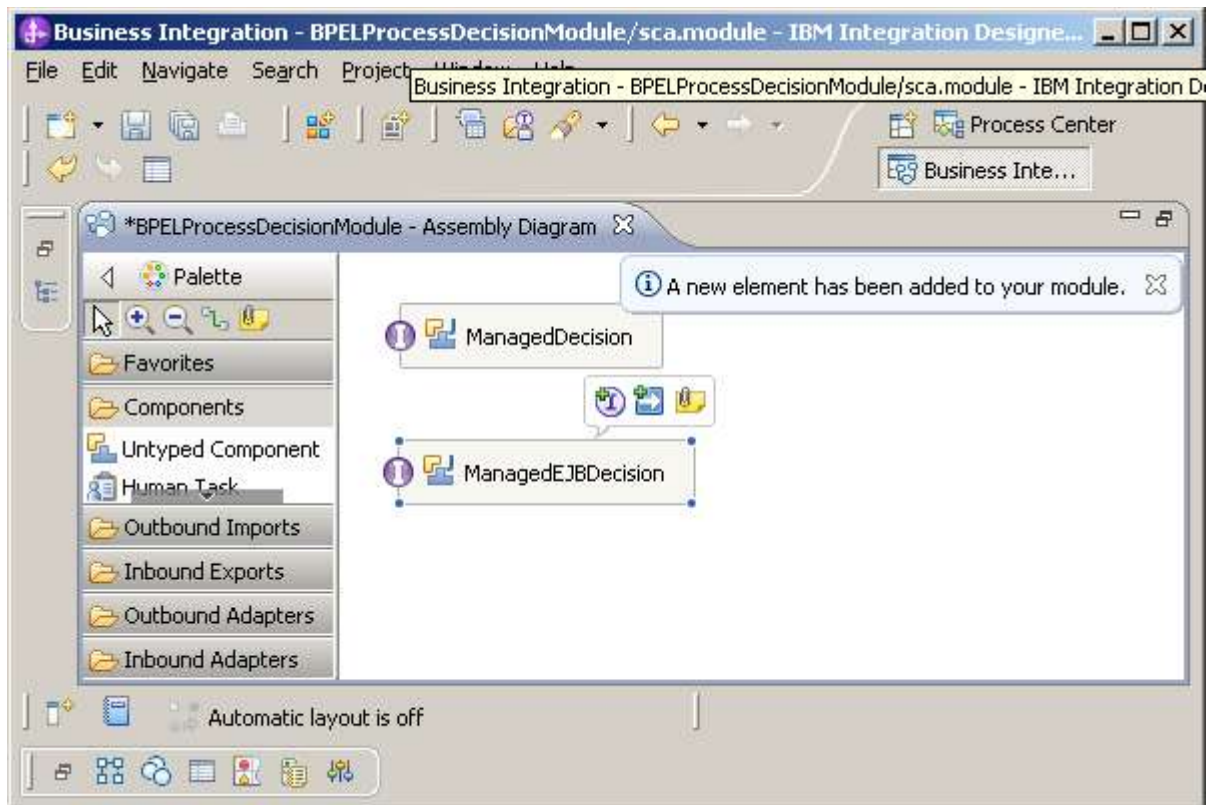
JNDI Name: ilog.rules.res.session.impl.ejb3.IlrStatelessSessionRemote

Remote URL: iiop://9.115.196.254:2809/cell/clusters/SingleCluster

? < Back Next > Finish Cancel

Click **Finish**.

The workspace should now show the Module Assembly Diagram with the new **ManagedEJBDecision** interface available.

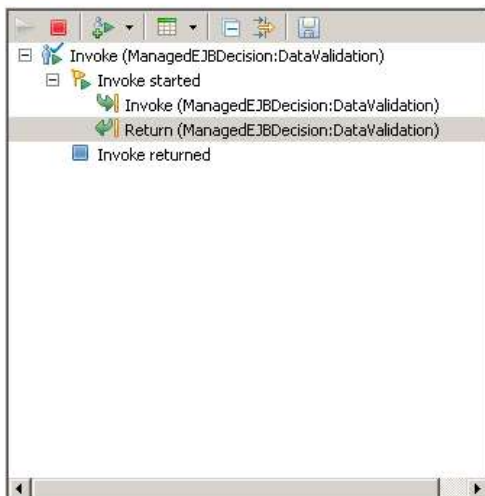


The EJB Session Bean that this connects to should have been installed as described in BPM Tutorial Overview so this component can be tested in the same manner as the POJO implementation in step 3.

Integration Test Client: BPELProcessDecisionModule_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...](#)



Detailed Properties

Module: [BPELProcessDecisionModule](#)
 Component: [ManagedEJBDecision](#)
 Interface: [ManagedEJBDecision](#)
 Operation: [DataValidation](#)

Return parameters:

Value Editor XML Source

Name	Type	Value
AntiLoc	boolean	false
DayTime	boolean	false
Make	string	...
Model	string	...
ModelYear	int	0
Value	double	0.0
Coverage	Coverage...	60
DataValidationR	Validation...	...
Validated	boolean	false
MainMessage	string	The number of accidents should not be negativ...