

IBM WEBSHERE 6.1 – LAB EXERCISE

Modeling business rules using WebSphere Business Modeler V6.1

What this exercise is about	1
Lab requirements	1
What you should be able to do	1
Introduction	2
Exercise instructions	2
Part 1: Creating the business rule template and business rules.....	3
Part 2: The implementation – exporting to WebSphere Integration Developer V6.1	15
What you did in this exercise	25
Solution instructions	26

What this exercise is about

The objective of this lab is to provide you with an understanding of the new feature in WebSphere Business Modeler V6.1 for modeling business rules. The business rules defined in WebSphere Business Modeler V6.1 can then be imported into WebSphere Integration Developer V6.1 where the business process implementation can be completed and tested.

This exercise will demonstrate how to create a business rule template and business rules in WebSphere Business Modeler V6.1 and the close integration with WebSphere Integration Developer V6.1.

Lab requirements

List of system and software required for the student to complete the lab.

- WebSphere Business Modeler V6.1
- WebSphere Integration Developer V6.1

What you should be able to do

At the end of this lab you should be able to:

- Create a local business rule template and business rule.
- Export the project and then import it into WebSphere Integration Developer V6.1 and understand the mapping between the WebSphere Business Modeler and WebSphere Integration Developer.

Introduction

Business rules are a collection of if-then-else statements that perform business logic at specified points in the business process. What makes business rules a powerful tool is that they can be managed externally to the running application and with the use of business rule templates and the business rules manager application, key values can be parameterized and changed by the business analyst at runtime.

The business rules created using WebSphere Business Modeler V6.1 are specifically designed to be used with WebSphere Integration Developer V6.1 and WebSphere Process Server V6.1

In this lab exercise you will begin by importing an existing V6.0.2 Modeler project that has a business rule defined, Because WebSphere Business Modeler V6.0.2 does not have first class support for business rules, the specification of the business rule is vague and incomplete.

Upon importing the modeler archive file, the project will be migrated to the V6.1 but the old style business rule will need to be converted to the new business rule task that is available with WebSphere Business Modeler V6.1

You will then create a business rule template and 2 business rules which use the template. This will demonstrate how to create the template parameters and use them for creating specific rule instances.

After completing the specification of the business rules in WebSphere Business Modeler V6.1 the model will be exported from WebSphere Business Modeler V6.1 using the WebSphere Integration Developer export type and then imported into WebSphere Integration Developer V6.1.

Using WebSphere Integration Developer V6.1 you will then locate and identify the business rule you created in modeler and understand how it is mapped to WebSphere Integration Developer V6.1

Exercise instructions

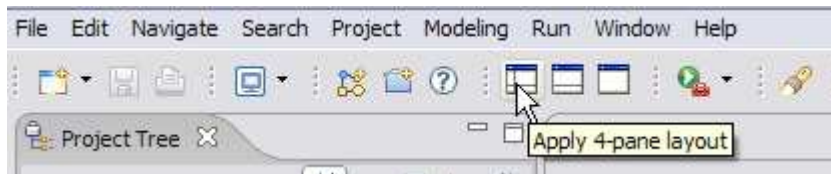
Some instructions in this lab are Windows[®] operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference variable	Windows location	AIX/UNIX location
<WAS_HOME>	C:\WebSphere60\AppServer	/usr/WebSphere60/AppServer /opt/WebSphere60/AppServer
<IRAD_HOME>	C:\Program Files\IBM\RSMP\6.0	
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp

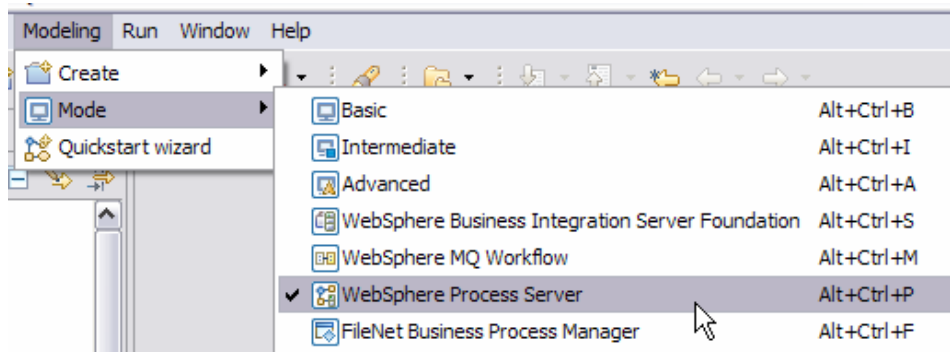
Note for Windows users: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFiles60\ with C:/LabFiles60/

Part 1: Creating the business rule template and business rules

- ___ 1. Download and extract the WBIV61_Labfiles_Modeler.zip from the IBM Education Assistant.
 - ___ a. It is under the additional resources where you downloaded this document from.
 - ___ b. Unzip it to the root of your drive.
- ___ 2. Start WebSphere Business Modeler V6.1 using a new workspace
 - ___ a. Call the workspace Business Rules
 - ___ b. Wait a few moments until the WebSphere Business Modeler opens.
 - ___ c. Close the welcome page
 - ___ d. Apply the 4-pane layout from the toolbar

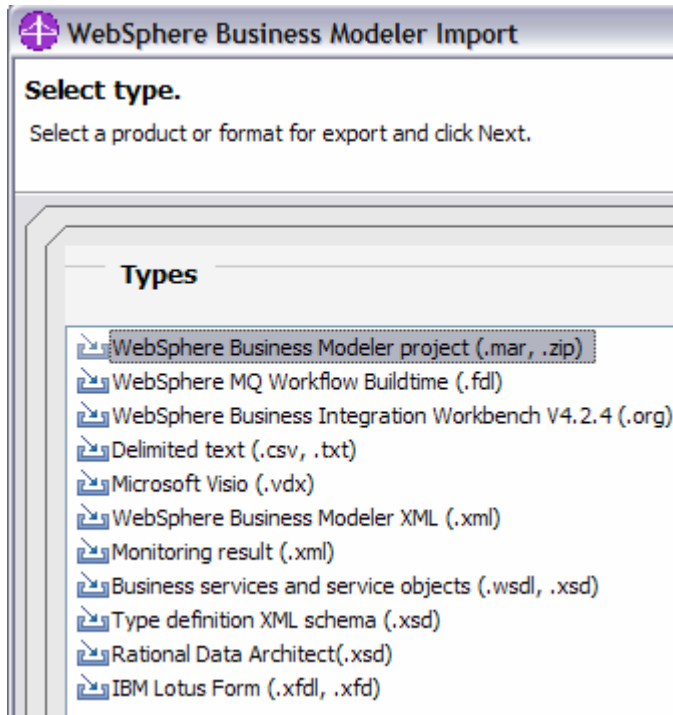


- ___ 3. Set the modeling mode to WebSphere Process Server



- ___ 4. Import the existing human resources project.
 - ___ a. Right click on in the background area of the project tree and select import.

__ b. Select the WebSphere Business Modeler project (.mar, .zip) import type and



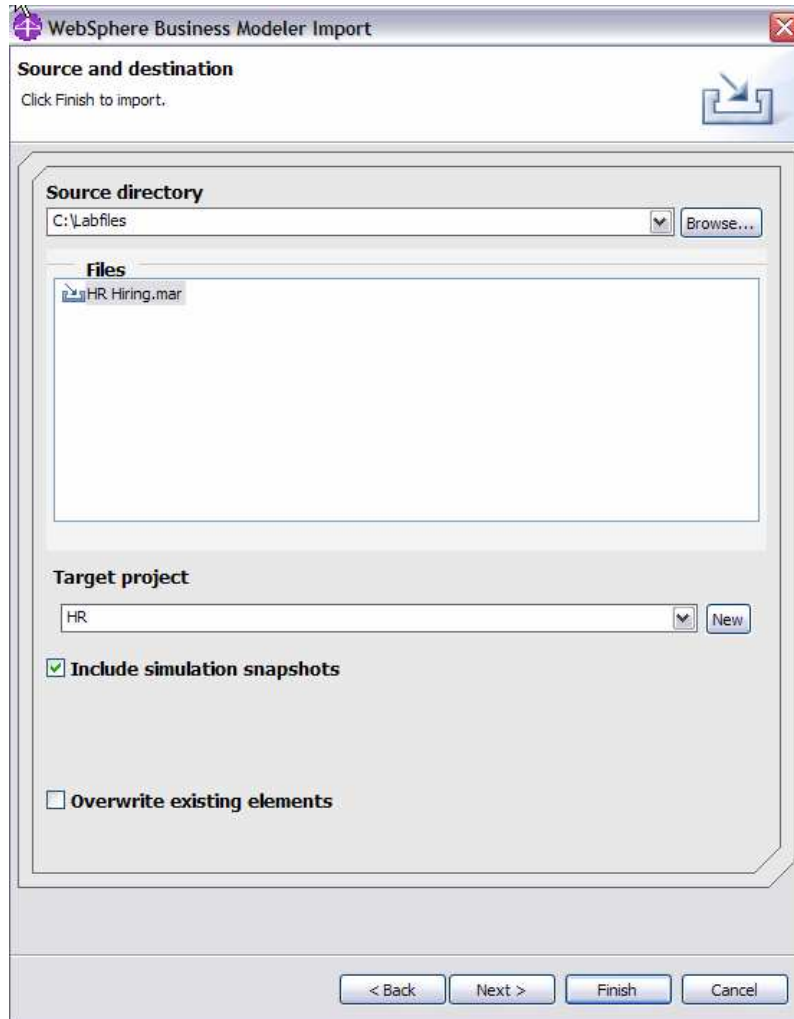
__ c. Browse to the source directory <LAB_FILES>\Modeler 6.1

__ d. Select the modeler archive, **HR Hiring.mar**

Note: This dialog is context sensitive and what is displayed depends on the file type selected. WebSphere Business Modeler V6.1 .mar files may or may not need to specify a target project whereas the 6.0.2 .mar files will.

__ e. Create a new target project using the **New** button.

- 1) Call the target project **HR**
- 2) After the target project has been created, press **Finish**.



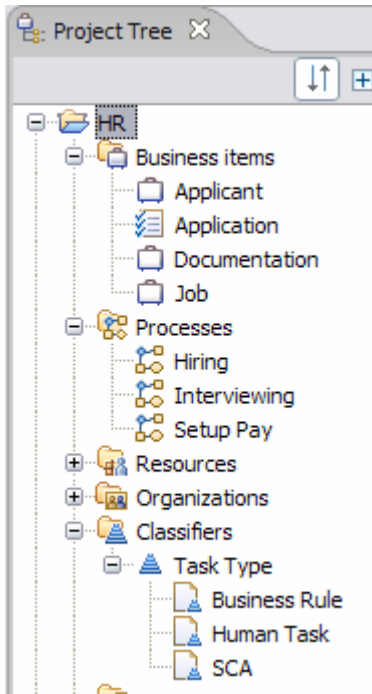
__ f. An informational dialog will be displayed.

1) Press OK



2) Ignore the warnings when complete.

5. Familiarize yourself with the model.

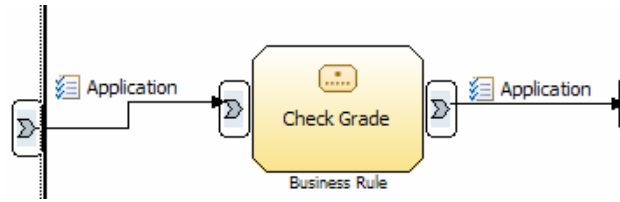


a. First notice the business processes. The overall business process that is captured in this model is the hiring of new employees. Interviewing the prospective employee is an integral part of the hiring process. In this exercise you will be working with the **Hiring** business process.

- 1) Open the Hiring business process and become familiar with it. Notice the labels and the overall flow.
- 2) Check Grade is a task designed to inspect key information in the incoming application, such as the salary requested and the job position and determine if the position being applied for is an executive position or not. If it is an executive position then a Boolean attribute in the application will be set to true. (*Job* is a field in the *Application* business item)

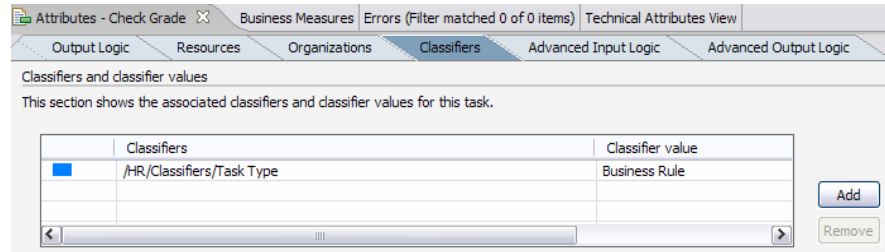
Name	Type
+ Applicant	Applicant
- Job	Job
Salary	Integer
Position	Text
IsExecutive	Boolean

- 3) Inspect the **Check Grade** task. This is the task that has been specified as a business rule. The areas of interest are the classifiers and the implementation tab in the technical attributes.



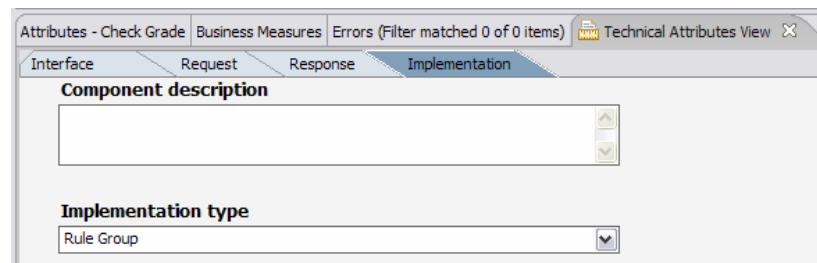
a) Select the task (single click) and review the tabbed pages in the bottom pane.

(1) Classifiers.



The classifier is used to help identify this particular task as a business rule. The main benefit is that the value of the classifier, *Business Rule*, can be displayed on the process diagram.

(2) Technical attributes – implementation.



The implementation type is a way to indicate to WebSphere Integration Developer V6 that this particular task is to be implemented as a business rule group. Since there is no further information about the business rules or rule group, WebSphere Integration Developer V6 can only create a skeleton implementation upon import. You'll see that with WebSphere Business Modeler V6.1, the story is much different.

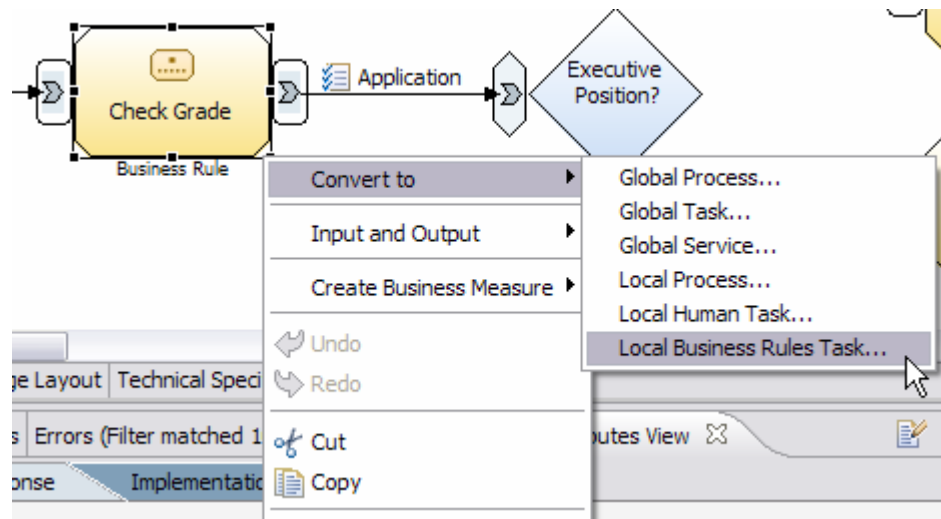
Note: If you do not see the technical attributes page, then you forgot to switch to WebSphere Process Server modeling mode. See step 2.

___ b. Notice the business items. The most significant one you'll be concerned with is the **Application**. The Application business item is passed through all of the tasks in the Hiring process.

___ c. Notice the classifiers. They are used to give you a visual cue as to the implementation type.

___ 6. Convert the V6.0.2 style business rule to a first class V6.1 business rule.

___ a. Right mouse click on the Check Grade task to get the pop-up menu.



and select convert to local business rules task.

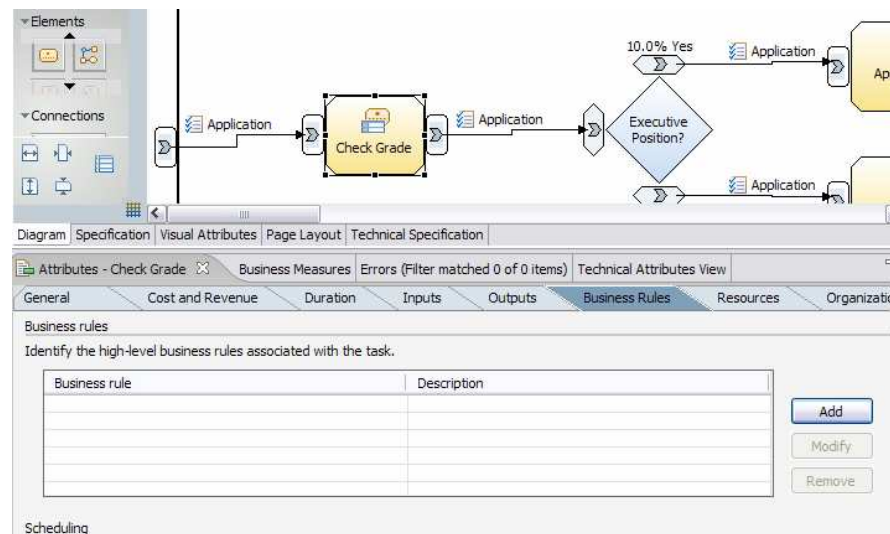
___ b. Save using ctrl-s

___ 7. Create the business rule (group) for the Check Grade business rule task.

___ a. Select the Check Grade business rule task

___ b. Select the Business Rules tab from the attributes specification page

___ c. Select the **Add** button.



___ d. Pressing the Add button will invoke the Define Business Rule dialog. Fill out the dialog as shown below.

___ e. Name and description: Check Grade Rules

___ f. Add a template definition. We will be making a parameterized rule template that will be used to create several specific rules that are the same with the exception of some of the key values.

Set the name to Check Grade by directly editing the cell in the table.

1) Do not fill out the rule condition or action yet. Since the rule condition will use parameters, it will be easier if the parameters already exist before creating the rule condition.

___ g. Add rule parameters.

1) To change the fields after adding a new parameter, select the field (single click) and you will be presented with a drop down box or a direct edit field, depending on the field.

___ h. Create the rule condition by clicking on the rule condition field and then the button in the field, to invoke the expression editor.

Important
To reuse rule conditions and actions or allow their parameter values to be changed in an application at runtime, create a rule template.

Rule Templates If-Then Rules

Definition
To specify rule conditions and actions, create any required rule parameters. Also ensure that the business rules task has inputs and outputs defined.

Template name	Rule condition	Rule action	Template description
Check Grade Template			Check the level of the po

Add Remove Move Up Move Down

Rule parameters
To change parameter values at runtime or add parameters to either the rule condition or action, specify rule parameters. Add constraint information as a description.

Parameter name	Type	Description
salary	Integer (long)	The amount the applicat expects to earn
position	Text	The position they are applying for.

Add Remove

___ i. Using the expression editor.

- 1) The expression editor is shown below after the expression has been constructed.
- 2) The area on top will remain blank until an expression is applied. Applying a second expression will use the **AND** logical connector by default. (It can be changed to **OR** if needed by directly editing the field.)
- 3) You will use a modeling artifact from the input as the first term, either salary or position and a modeling artifact, a parameter, for the second term, with the appropriate connector in between.

Expression Composer - Subexpression

First term:
Modeling artifact

First term details:
Modeling artifact:

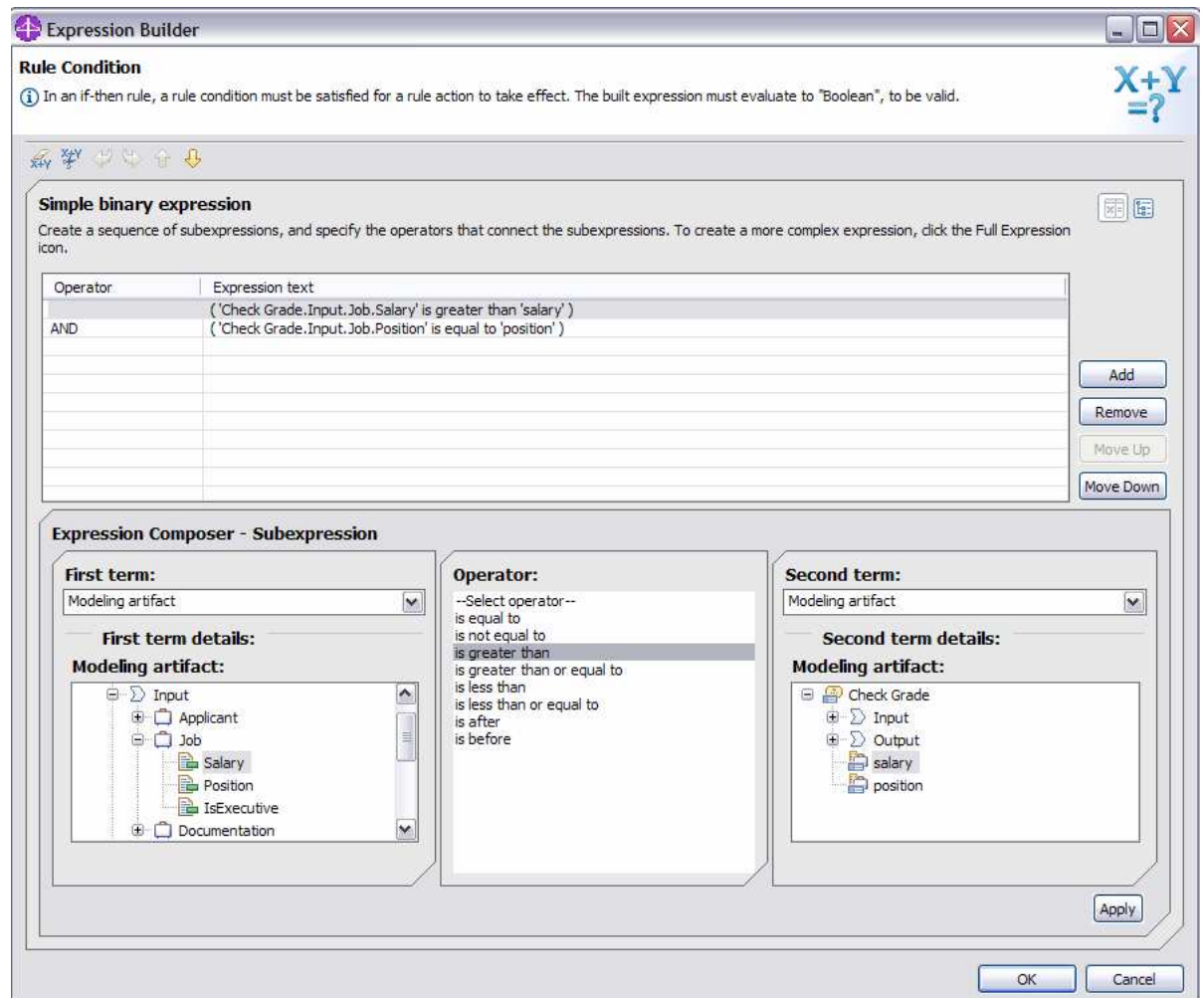
- Input
 - Applicant
 - Job
 - Salary
 - Position
 - IsExecutive
 - Documentation

Operator:
--Select operator--
is equal to
is not equal to

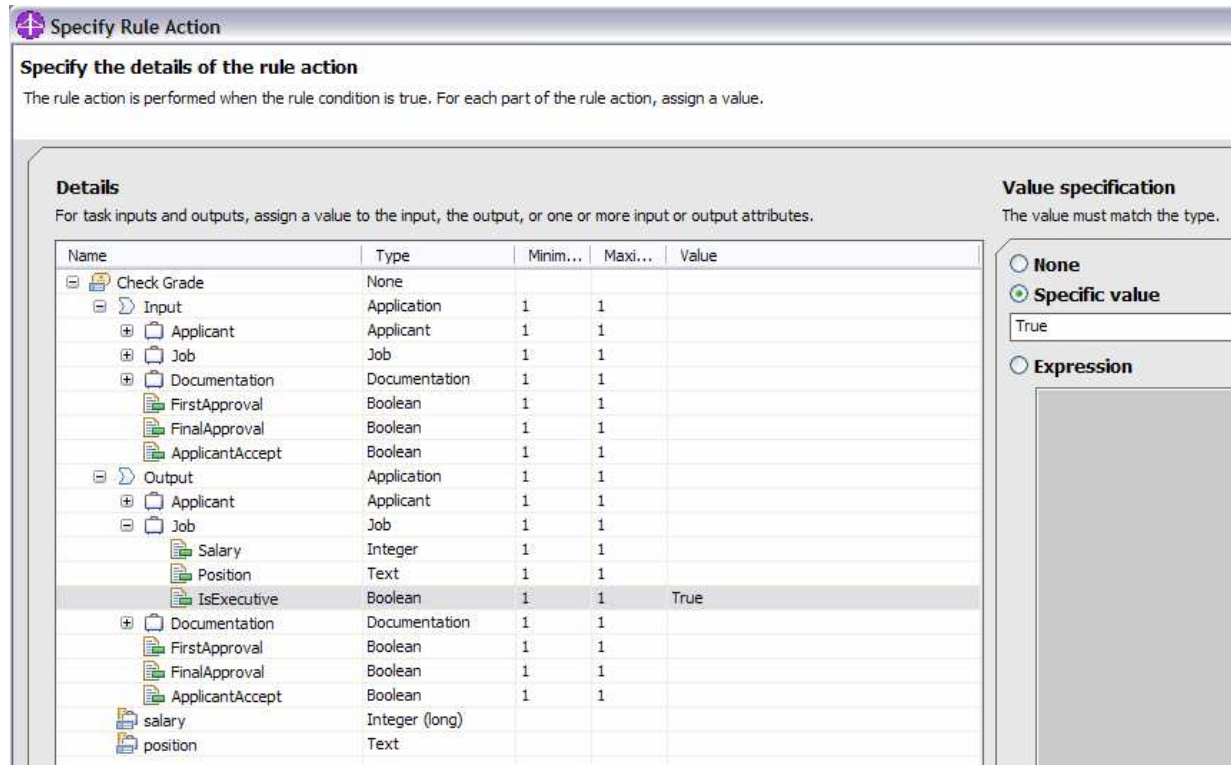
Second term:
Modeling artifact

Second term details:
Modeling artifact:

- Check Grade
 - Input
 - Output
 - salary
 - position

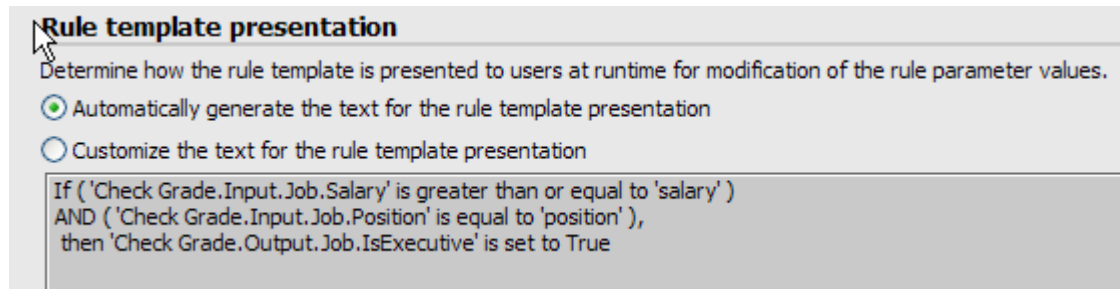


- 4) Start building your expression by pressing the add button.
 - a) **'Check Grade.Input.Job.Salary' is greater than or equal to 'Check Grade.salary'**.
 - b) The lower case salary is the variable (parameter).
 - 5) Build the first expression and then press add again to build the second expression.
 - a) **'Check Grade.Input.Job.Position' is greater than or equal to 'Check Grade.position'**.
 - b) The lower case position is the variable (parameter).
 - 6) When the expression is complete, press **OK**.
- ___ j. Create the rule action by clicking on the rule action field and then the button in the field to invoke the action details
- 1) Expand **job** attribute associated with the output to expose the **IsExecutive** attribute.
 - 2) In the value specification pane on the right, select a specific value of true.



3) Press **OK**.

___ k. Inspect the rule template presentation.

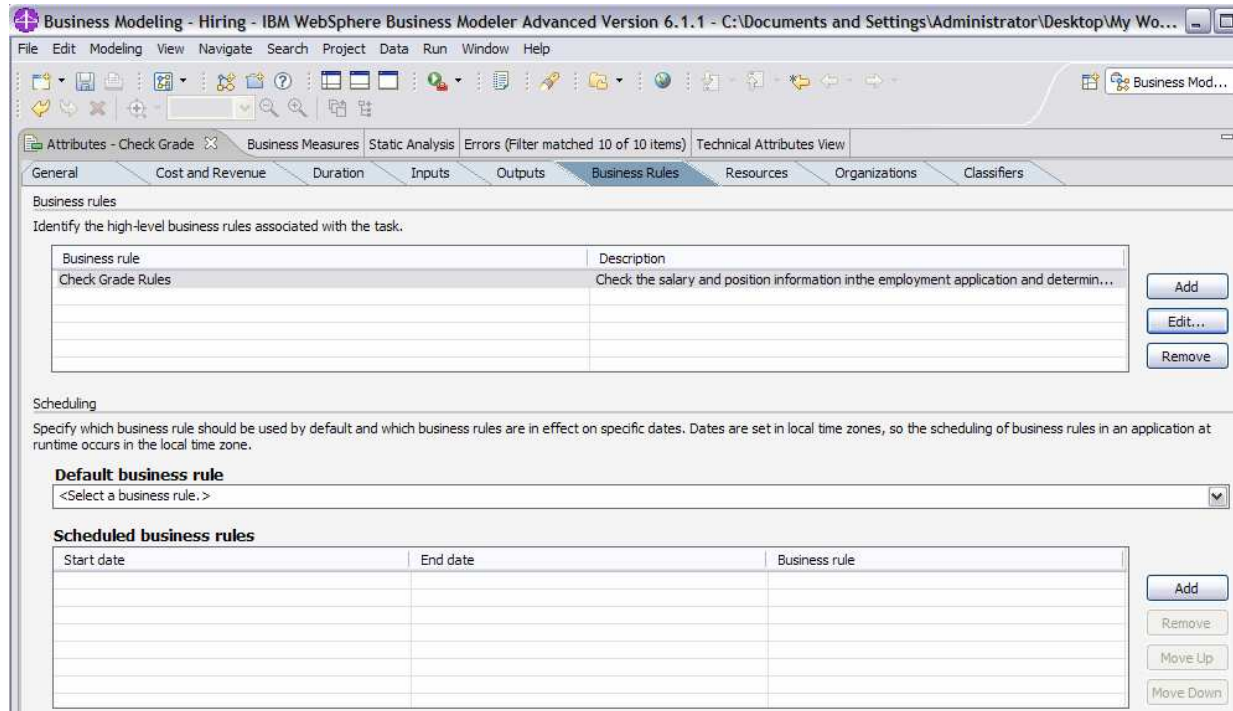


1) If you like, customize the presentation to be something that you think is more suited to a business analyst rather than a programmer. The information provided here will be displayed when the business analyst uses the Business Rules Manager to edit a rule at run time.

___ l. Click **OK** to save your work. This will take you to the business rules page.

___ m. Create specific business rules using the Check Grade business rule template.

1) From the business rules page, make sure that the Check Grade rule is selected then select the **Edit** button.



2) In the Define Business Rule dialog, select the **If-Then Rules** tab.

a) You are going to create 2 if-then rules based on of the Check Grade business rule template.

(1) The CTO rule, salary >= \$150,000 and position = CTO

(2) The Release Manager rule, salary >= \$120,000 and position = RM

b) Click **Add Rule**.

c) Fill in the name, CTO Rule and then select the template. You'll get a warning.



d) Press OK

e) The fill in the values in the rule parameter values section.

Rule parameter values

Specify the value of each parameter used in the rule condition and action.

Parameter name	Type	Value	Description
salary	Integer (long)	150000	How much do they expect to make
position	Text	CTO	What kind of position are they looking for

f) Repeat the procedure for the release manager rule using \$120,000 for the salary and RM for the position.

Rule Templates **If-Then Rules**

If-Then Rules

Select a rule template or define an if-then rule. To create rule conditions and actions, ensure that the business rules task has inputs and outputs defined.

Rule name	Template name	Rule condition	Rule action	Rule description
CTO	Check Grade Template	('Check Grade.Input.Job.Salary' is greater than 'salary') AND ('Check Grade.Input.Job.Position' is equal to 'position')	'Check Grade.Output.Job.IsExecutive' is set to True	<Type the description here. >
Release Manager	Check Grade Template	('Check Grade.Input.Job.Salary' is greater than 'salary') AND ('Check Grade.Input.Job.Position' is equal to 'position')	'Check Grade.Output.Job.IsExecutive' is set to True	<Type the description here. >

Rule parameter values

Specify the value of each parameter used in the rule condition and action.

Parameter name	Type	Value	Description
salary	Integer (long)	120000	How much do they expect to make
position	Text	RM	What kind of position are they looking for

g) Press **OK**.

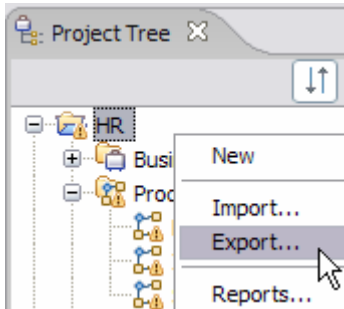
h) Ctrl-s to save your work.

Part 2: The implementation – exporting to WebSphere Integration Developer V6.1

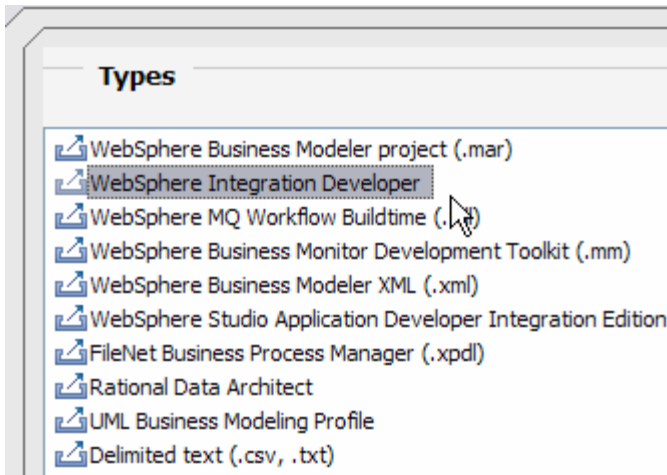
Capturing the details of the business rules in WebSphere Business Modeler may be all that is required. Chances are that the real goal is to create a model that will be implemented in WebSphere Integration Developer V6.1 and run in WebSphere Process Server V6.1.

To do this the next step is to export the model you've just created and then import it into WebSphere Integration Developer V6.1. Part 2 of this exercise will take you through those steps.

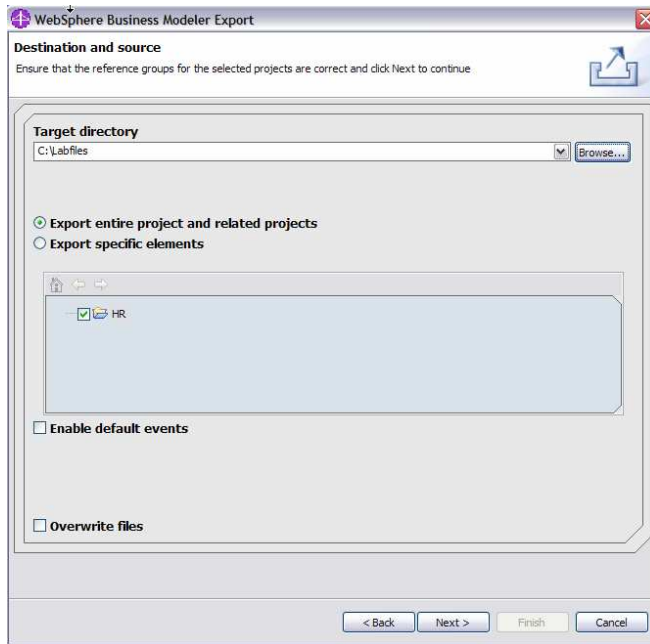
- ___ 1. Right mouse click on the HR project to get the pop-up menu and select export.



- ___ 2. Export the HR modeling project using the WebSphere Integration Developer export type.

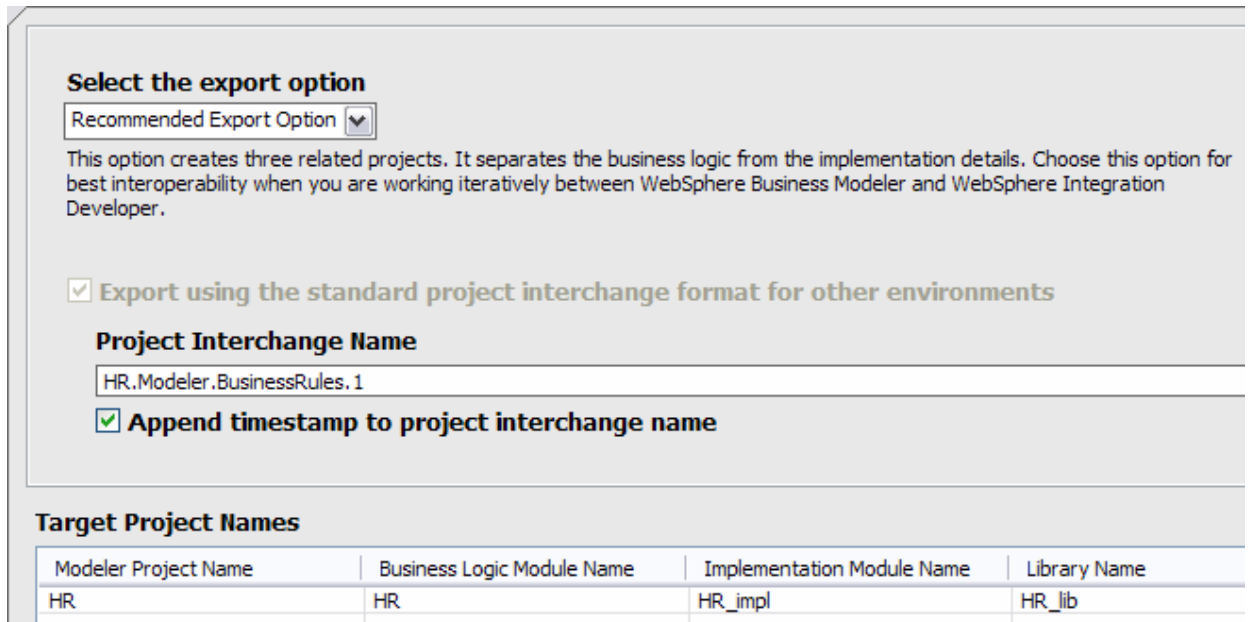


___ a. Set the target directory = <LAB_FILES>



___ b. Press **Next**.

___ c. Use the **recommended export option**. This will be selected by default. When this option is used three projects will be created for you in WebSphere Integration Developer V6.1 based on the separation of concerns principle discussed in the lecture.



1) **Business Logic Module:** This project will contain all of the generated BPEL and any concrete processing elements such as the business rules and human tasks that represent the business model.

a) BPEL, inline TEL files, WSDL files related to the BPEL, exports and imports.

- 2) **Implementation Module:** This project will contain the implementations for the Web services or other tasks for which there is no concrete implementation. The implementations will be linked to the BPEL activities in the business logic module through SCA import/exports.

This separation between the Model and the Implementation will help preserve the implementations when there are changes in the model.

Note: A regular task in modeler with an implementation type set in the technical attributes will have a skeleton of the specified type created in the integration module project.

- 3) **Library Module:** This project contains all interfaces and data definitions shared by the business logic and the implementation modules.

a) Artifacts included in the library are BOs and WSDL interfaces for all elements (global and local).

__ d. Export using the standard project interchange format

__ e. For the Project Interchange Name: **HR.Model.BusinessRules.1**

__ f. Append the timestamp.

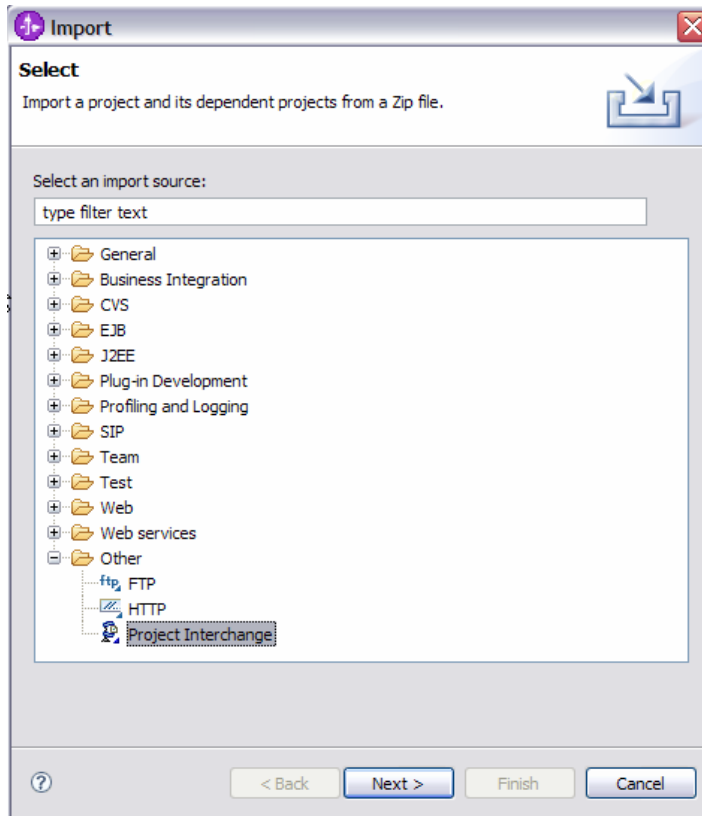
__ g. Press **Finish**.

- 1) There will be a warning about the lack of a correlation id. This is ok. The correlations is something the integration developer will be able to take care of.

___ 3. Start WebSphere Integration Developer V6.1 using a new workspace.

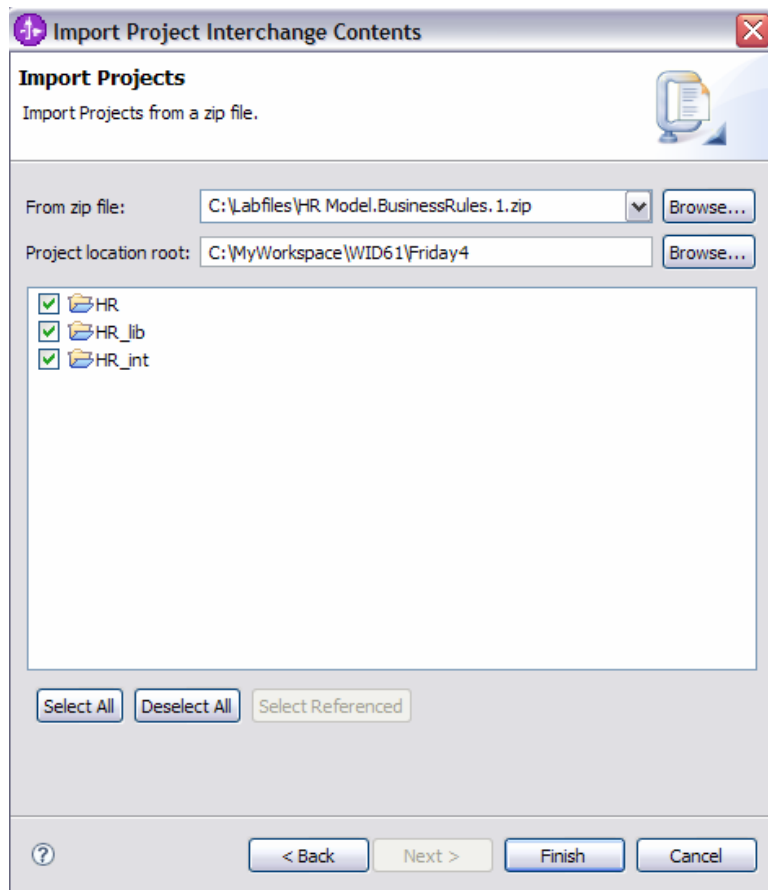
___ 4. Import the HR.Model.BusinessRules.1.zip

__ a. Locate the Project Interchange import type in the "Other" folder.



__ b. Press **Next**.

___ c. Notice the three types of projects. Ensure that all three are selected.



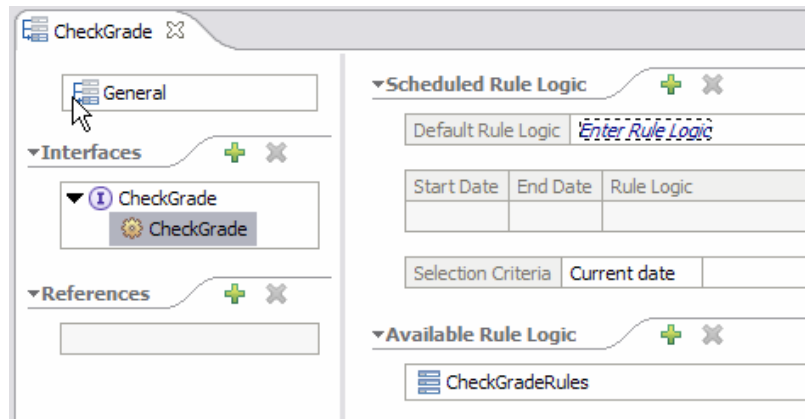
___ d. Press **Finish**.

___ e. Wait for the projects to finish building. The progress indicator is at the bottom right of the window in the message bar.

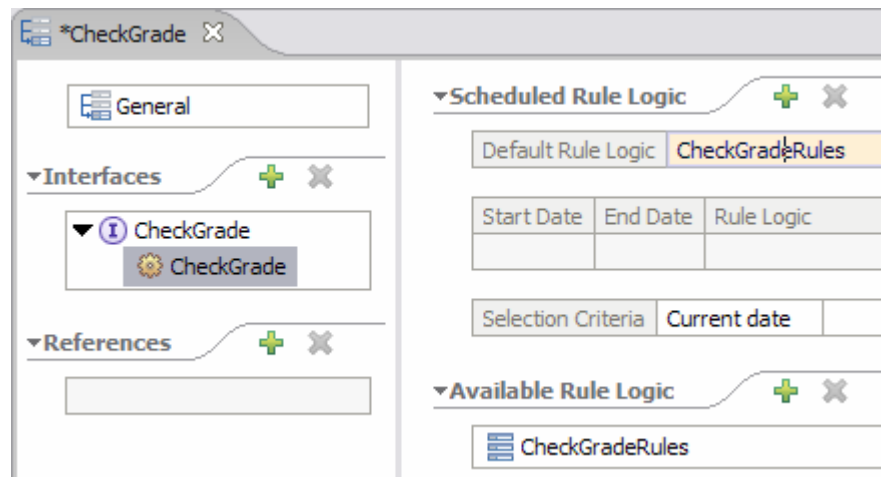
___ f. There should only be two errors in the problems list. The error regarding the missing correlation set can be ignored. It is not in the business process you're concerned with at this time.

The error regarding the business rule is another matter. This error is because you did not set a default business rule. This is easy to correct at this stage.

- 1) Double click on the error and it will open the Check Grade rule group.



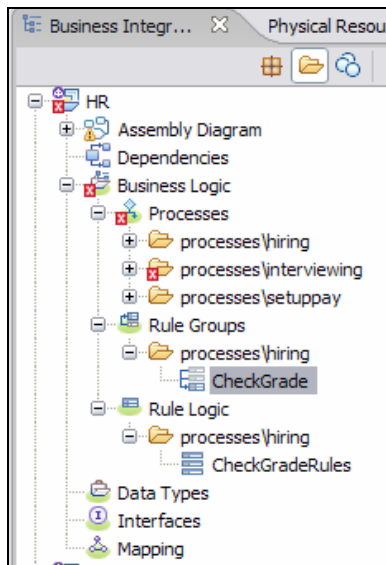
Select the CheckGrade gear on the left and then the *Enter Rule Logic* on the right. From the selection list, select CheckGradeRules.



2) Ctrl-s to save and the error will go away.

- ____ 5. Browse the projects keeping in mind that the focus here is on the business rules in the **Hiring** business process.

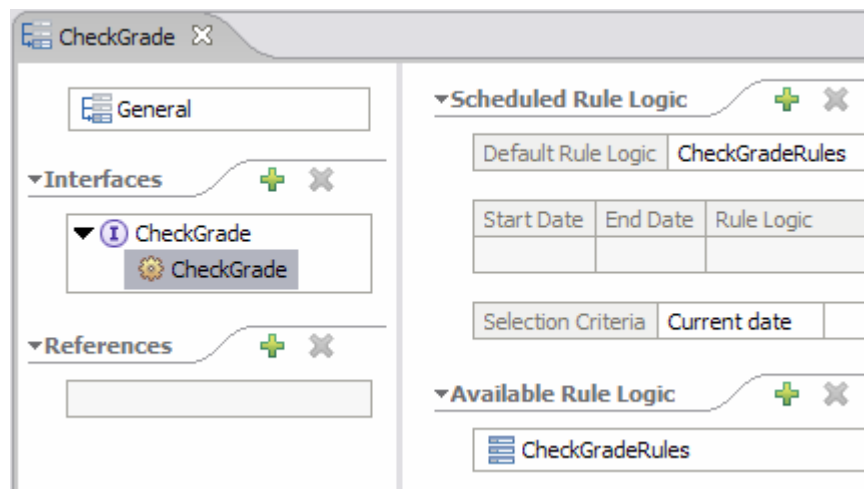
___ a. HR (business logic module) – expand the *Business Logic* folder



___ b. Notice that there are 2 components related to Business Rules.

1) The CheckGrade Rule Groups

a) The rule group comes from the Business Rule Task in Modeler. It is the container for the rules and has a pointer to the CheckGradeRules rule logic.



2) The CheckGradeRules Rule Logic.

a) The rule logic comes from the definitions of the business rule template and if-then rules and actions.

b) Open the CheckGradeRules. The information here should be very familiar. It is everything you specified in the WebSphere Business Modeler model.

CheckGrade
CheckGradeRules X

▼ Rule Set

Name	CheckGradeRules	Display Name	Check Grade Rules
------	-----------------	--------------	-------------------

▼ Interface

Interface	CheckGrade		
Operation	InputCriterion		
Input	Input		Application
Output	Output		Application

▼ Variables

Name	Type

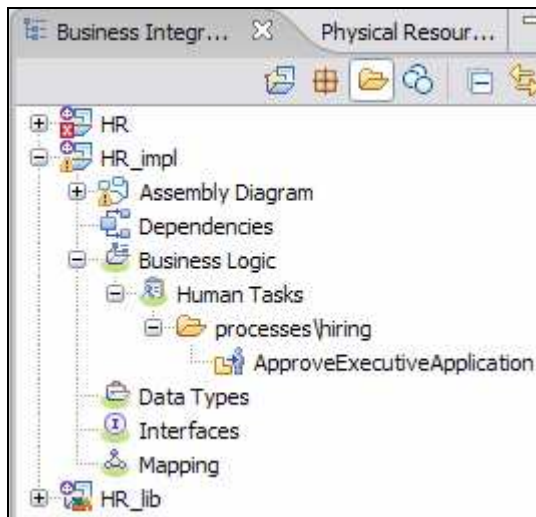
▼ Rules

Name	CTORule
Template	CheckGradeTemplate
Presentation	If (Check Grade.Input.Job.Salary is greater than or equal to 150000) AND (Check Grade.Input.Job.Position is equal to CTO), then Check Grade.Output.Job.IsExecutive is set to True
Name	ReleaseManagerRule
Template	CheckGradeTemplate
Presentation	If (Check Grade.Input.Job.Salary is greater than or equal to 120000) AND (Check Grade.Input.Job.Position is equal to RM), then Check Grade.Output.Job.IsExecutive is set to True

▼ Templates

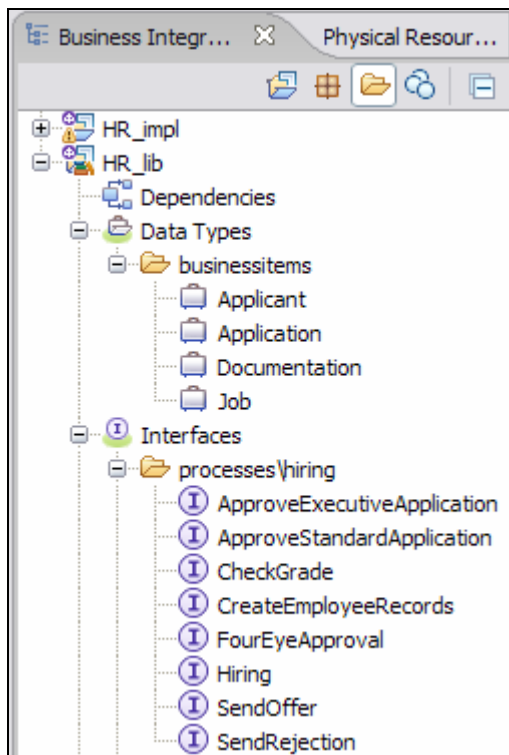
Name	CheckGradeTemplate			
Presentation	If (Check Grade.Input.Job.Salary is greater than or equal to salary) AND (Check Grade.Input.Job.Position is equal to position), then Check Grade.Output.Job.IsExecutive is set to True			
Description	Check the level of the position			
Parameters	Name	Type	Constraint	Description
	salary	long	None	The amount the applicat expects to earn
	position	string	None	The position they are applying for.
If	all of the following are true <ul style="list-style-type: none"> ● Input.Job.Salary >= salary ● Input.Job.Position == position 			
Then	Output.Job.IsExecutive = true			

__ c. HR_impl (implementation module) – expand the Business Logic folder.



- 1) There is one component associated with the **Hiring** business process and it's a human task. It's worth mentioning how this got here. If you go back to modeler you'll see that the **ApproveExecutiveApplication** task is labeled (from the classifier) as a human task but is not one of the new V6.1 human task process elements. The skeleton for the human task component was created because the implementation type, in the technical attributes, was set to human task.

__ d. HR_lib (library module project) – expand the data types and interfaces to see what is there for the hiring business process.

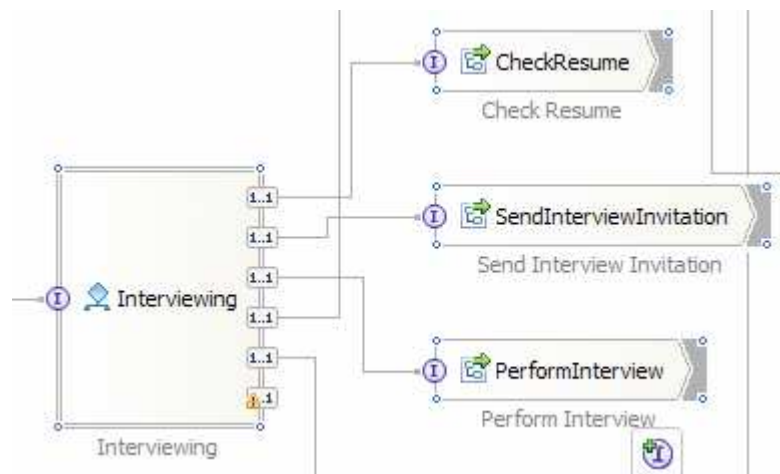


- 1) There are no surprises here, which is good. This is everything you'd expect to see. You can verify the data types and interfaces if you like.
- 2) Notice that all of the interfaces are 2-way except for the Hiring Interface.

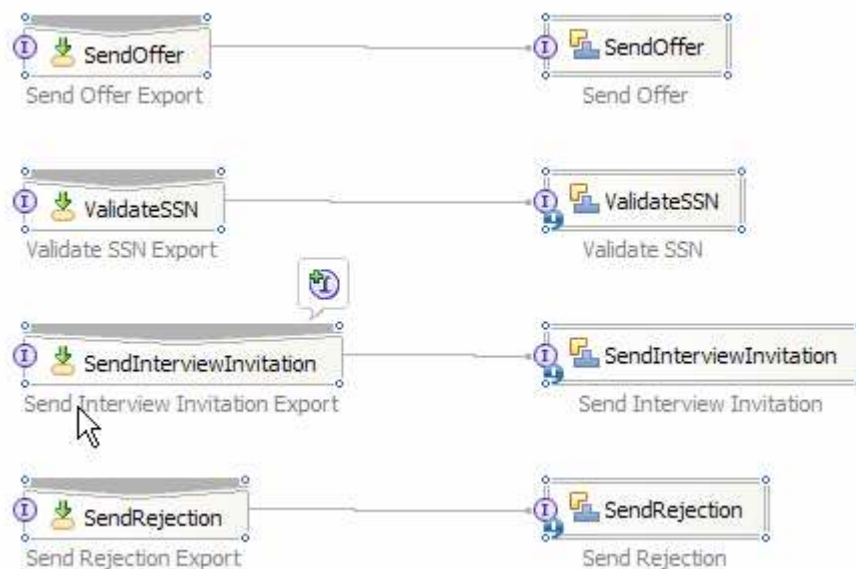
___ e. The assembly diagrams.

- 1) For a complete understanding of the relationships between the Module project and the Integration projects you'll need to study the assembly diagrams.
- 2) Notice that the general modeler tasks get mapped to BPEL invokes in the HR module and that in the assembly diagram, they are all linked to SCA imports. In the assembly diagram for the HR_impl module there are corresponding imports and interfaces. In this way the implementations for the services (tasks) are isolated from the changes in the model.

a) HR (assembly snippet)



b) HR_impl (assembly snippet)



What you did in this exercise

In this exercise you created a business rule in WebSphere Business Modeler V6.1 and exported it to WebSphere Integration Developer V6.1. In WebSphere Integration Developer V6.1 you inspected the results to gain an understanding of how the constructs map from one environment to the other.

Solution instructions

The solution is in 2 parts, the WebSphere Business Modeler part and the WebSphere Integration Developer part.

The WebSphere Business Modeler part is in the file:

\\abfiles\Modeler\solutions\HR.Model.BusinessRules.1.mar

The solution for WebSphere Integration Developer is in the file:

\\abfiles\Modeler\solutions\HR.Model.BusinessRules.1.zip