

IBM WebSphere® Process Integration 6.0 - Lab Exercise

AutoClean – Business Rules and Selectors

What this exercise is about	1
Lab Requirements	1
What you should be able to do	2
Introduction	2
Exercise Instructions	3
Part 1: Initialize the Workspace for this Lab Exercise.....	5
Part 2: Import Simple Java Implementation (a POJO).....	9
Part 3: Create a Business Rule Group.....	14
Part 4: Create a Business Rule Set	16
Part 5: Create a Selector.....	22
Part 6: Create an Export on Assembly Editor	25
Part 7: Publish the AutoCleanApp	28
Part 8: Update Business Rules using the Business Rule Editor Web tool.	30
Part 9: Test the Business Rule using Component Test.....	35
What you did in this exercise	38
Solution Instructions.....	39
Task: Adding Remote Server to WebSphere Integration Developer Test Environment.....	40

What this exercise is about

This exercise is about learning the basics of Business Rules and Selectors while learning how to use WebSphere Integration Developer V6 and the WebSphere Process Server V6.

Lab Requirements

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

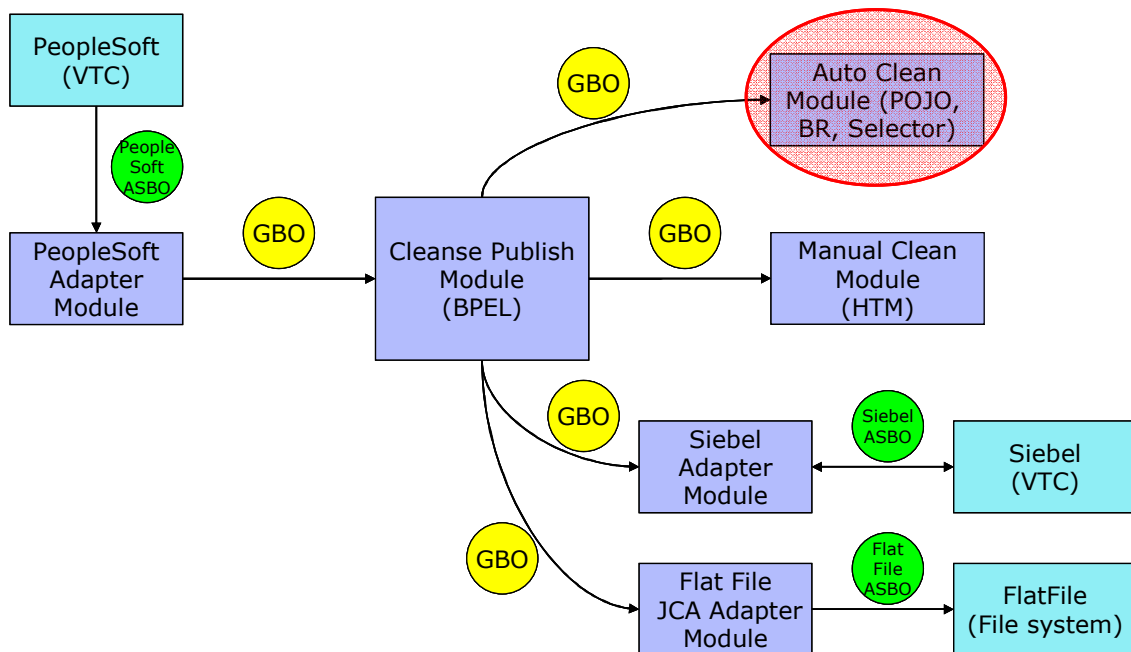
- WebSphere Integration Developer V6.0.1 installed
- WebSphere Process Server V6.0 test environment installed
- Sample code in the directory C:\Labfiles60 (Windows®) or /tmp/LabFiles60 (Linux®)

What you should be able to do

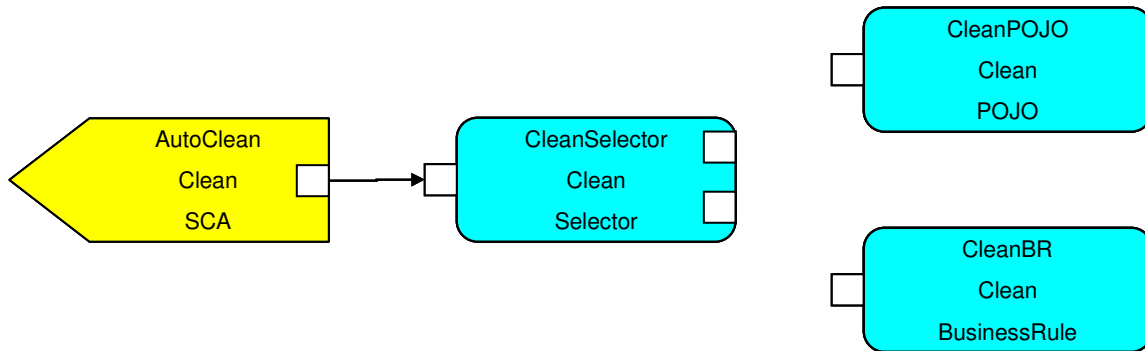
- Know how to import files into your integration projects
- Create and use Business Rule Groups, Business Rules, and Selectors
- Install and use the BRManager Web tool

Introduction

In the diagram below, the CleansePublish Module in the middle acts as a Client. It sends a GBO or Generic Business Object to have the data in the GBO to be cleaned up for whoever is viewing that data by the AutoClean Module. The AutoClean Module is connected to the CleansePublish Module by way of an Export/Import. So in the lower diagram, the GBO is passed through a Selector called CleanSelector to determine which Target Implementation to use, either the CleanPOJO or the CleanBR business rule. This choice is determined by a date; either the current date or a date variable mined using Java™ or Xpath parameters.



Description of the Module



Exercise Instructions

Some instructions in this lab might be specific for Windows platforms. If you run the lab on a platform other than Windows, you will need to run the appropriate commands, and use appropriate files (for example .sh in place of .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references as follows:

Reference Variable	Windows Location	Linux Location
<WID_HOME>	C:\Program Files\IBM\WebSphere\ID\6.0	/opt/IBM/WebSphere/ID/6.0
<WPS_HOME>	<WID_HOME>\runtimes\bi_v6	<WID_HOME>/runtimes/bi_v6
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<WORKSPACE>	C:\Labfiles60\eXchange\AutoClean\workspace	/tmp/Labfiles60/eXchange/AutoClean/workspace
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles60\AutoClean\Solution	/tmp/AutoClean/Solution

Windows users: When directory locations are passed as parameters to a Java program such as wsadmin, you must replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles60\ would be replaced by C:/LabFiles60/.

Note that the previous table is relative to where you are running WID. The following table is related to where you are running remote test environment:

Reference Variable	Example: Remote Windows test server location	Example: Remote z/OS [®] test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	cl1sr01	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/cl1cell/AppServerNode1	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<BOOTSTRAP_PORT>	2809	2809	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	cl1admin	
<PASSWORD>	N/A	fr1day	

Instructions for using a remote testing environment, such as z/OS, AIX[®] or Solaris, can be found at the end of this document, in the section "[Task: Adding Remote Server to WebSphere Integration Developer Test Environment](#)".

Part 1: Initialize the Workspace for this Lab Exercise

___ 1. Follow the directions below to initialize the Workspace using the following values:

<WORKSPACE>

C:\Labfiles60\exchange\AutoClean\workspace

<PROJECT_INTERCHANGE>

C:\Labfiles60\exchange\CleansePublishLibrary\import\CleansePublishLibrary_Pl.zip

<MODULE>

AutoClean

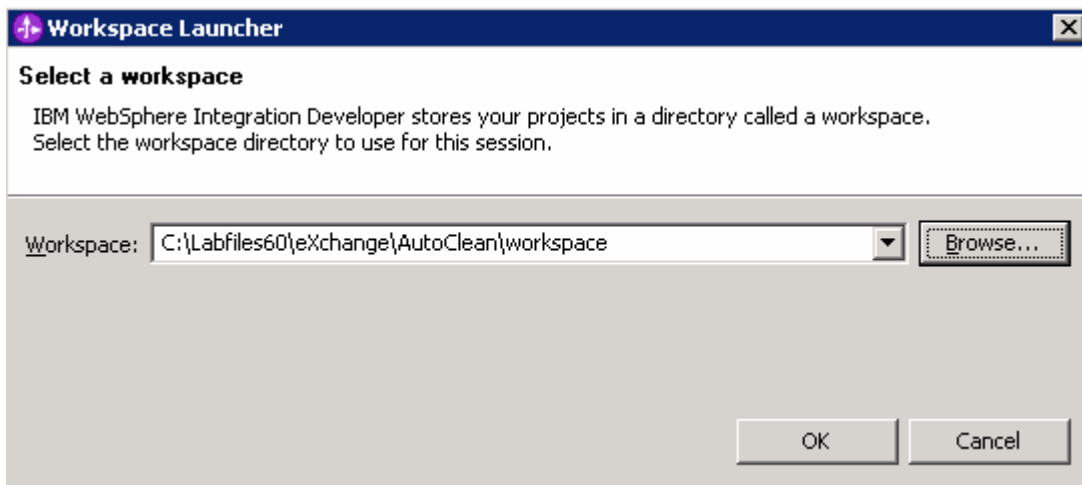
<DEPENDENT_LIBRARIES>

CleansePublishLibrary

___ 2. Start WebSphere Integration Developer V6.0.1 with a new workspace located at **<WORKSPACE>**.

___ a. From Windows Explorer, navigate to the **<WID_HOME>** directory and double click on wid.exe.

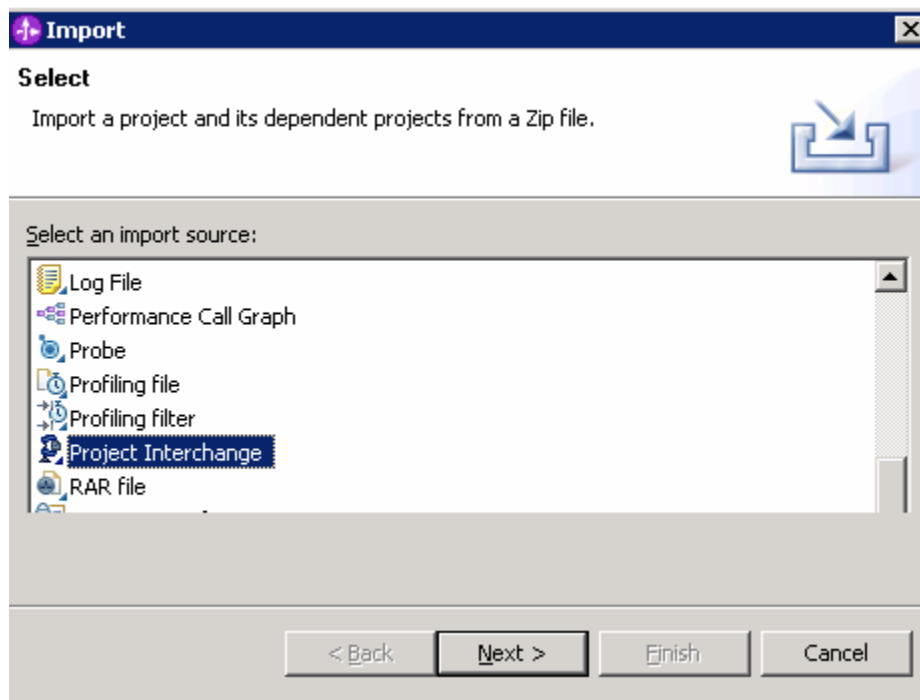
___ b. When prompted for workspace name, enter the value provided by the **<WORKSPACE>** variable for this lab and click **OK**.



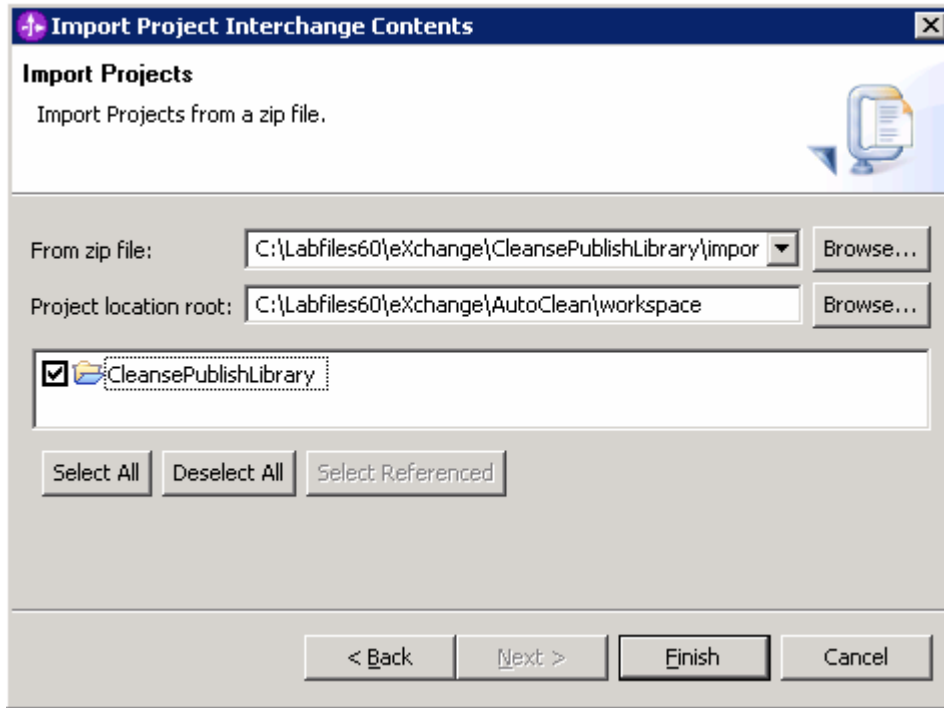
- ___ c. When WebSphere Integration Developer V6.0.1 opens, close the **Welcome page** by clicking on the Go to the workbench icon (bent over arrow at top-right).



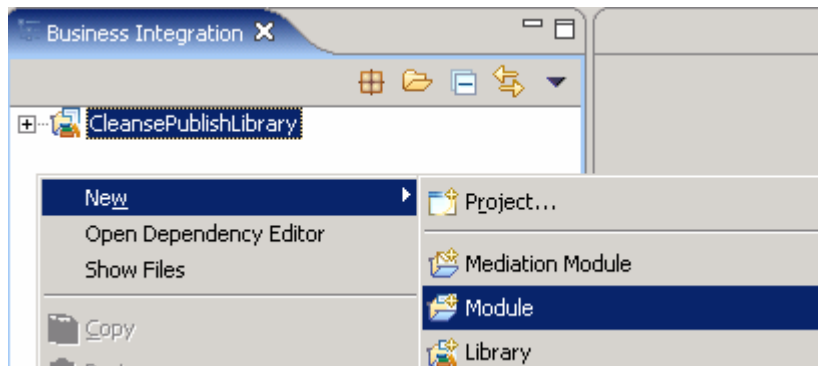
- ___ d. Ensure you are in the **Business Integration** perspective (See the task [Switching Perspectives within a workspace](#)).
- ___ 3. If this lab requires you to import a project interchange file, setup the required libraries and modules for this lab by importing the project interchange file <PROJECT_INTERCHANGE>.
- ___ a. From the menu bar, select **File > Import...**
 - ___ b. In the Import dialog, scroll down and select **Project Interchange**.



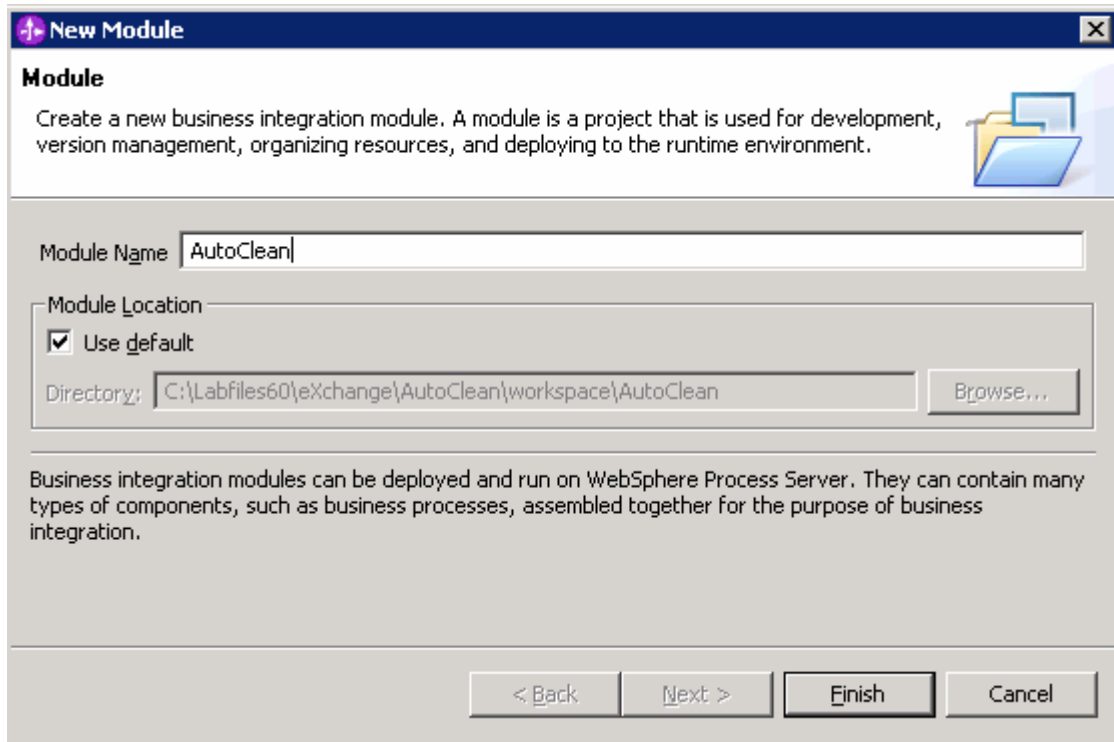
- ___ c. Click **Next**.
- ___ d. In the Import Projects dialog, initialize the From zip file: field to <PROJECT_INTERCHANGE>.
- ___ e. Click the **Select All** button.



- ___ f. Click **Finish**.
- ___ 4. If this lab requires that you create a Business Integration module called <MODULE>, complete these steps:
 - ___ a. Right click on the background of the Business Integration view to access the pop-up menu.
 - ___ b. Select **New > Module**.



__ c. In the New Module dialog, enter **<MODULE>** for the Module Name.

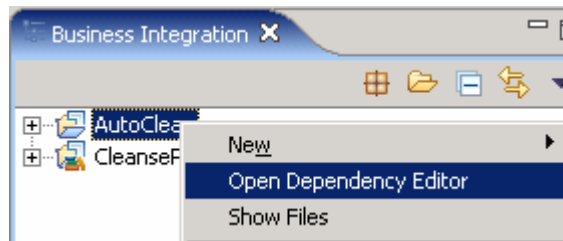


__ d. Click **Finish**.

___ 5. If this lab requires that **<MODULE>** needs any **<DEPENDENT_LIBRARIES>**, complete these steps.

__ a. In the Business Integration view, right click on the **<MODULE>** you just created to access the pop-up menu.

__ b. Select **Open Dependency Editor**.



__ c. In the Dependency Editor, click the **Add...** button.

__ d. In the Library Selection dialog, select from the list the **<DEPENDENT_LIBRARIES>**.

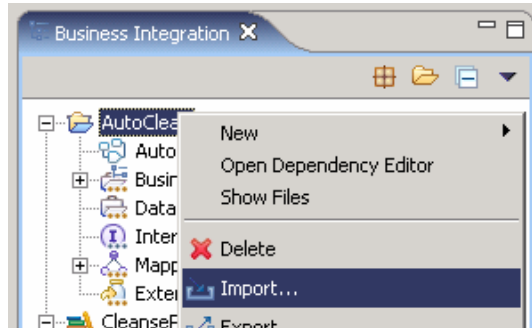
__ e. Click **OK**.

__ f. Press **Ctrl+S** to save the dependencies for this module.

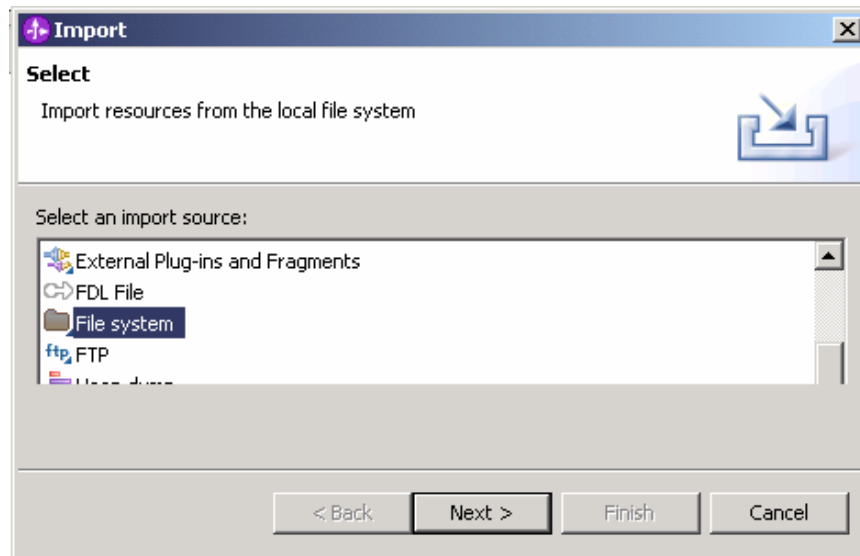
Part 2: Import Simple Java Implementation (a POJO)

During the first iteration of the cleanse process a simple Java implementation will be created to handle the cleanse process. This means you need a Java implementation to start the process you are going to test. This process involves a Java implementation sending a message through a selector, where the selector chooses to use a POJO (Plain Old Java Object) or a Business Rule given a certain date.

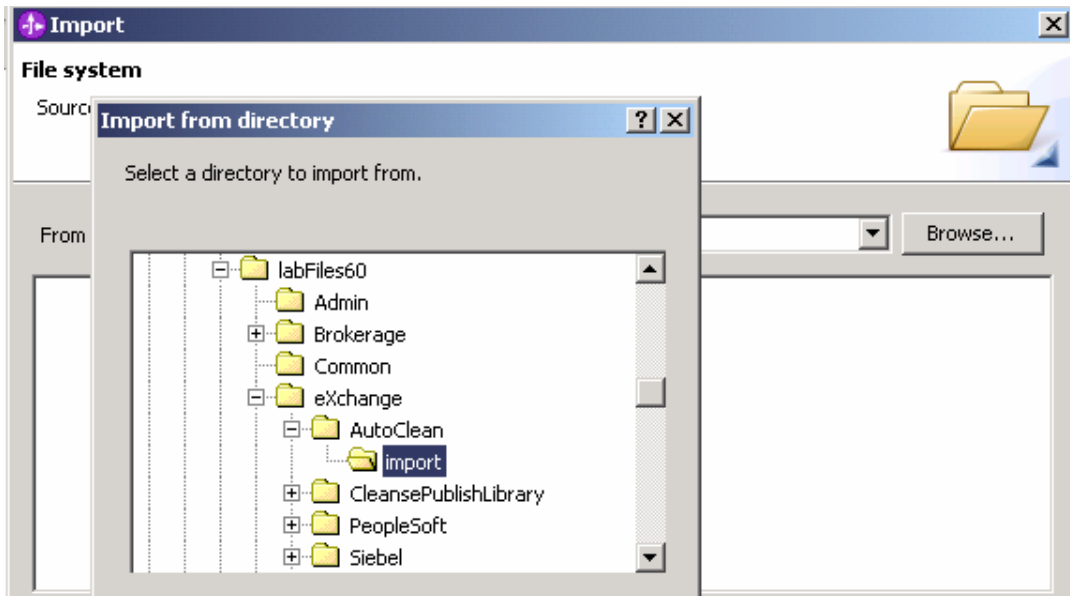
- ___ 1. Import the POJO (Java implementation).
 - ___ a. Right-click the **AutoClean** folder in the Business Integration view, select **Import...**



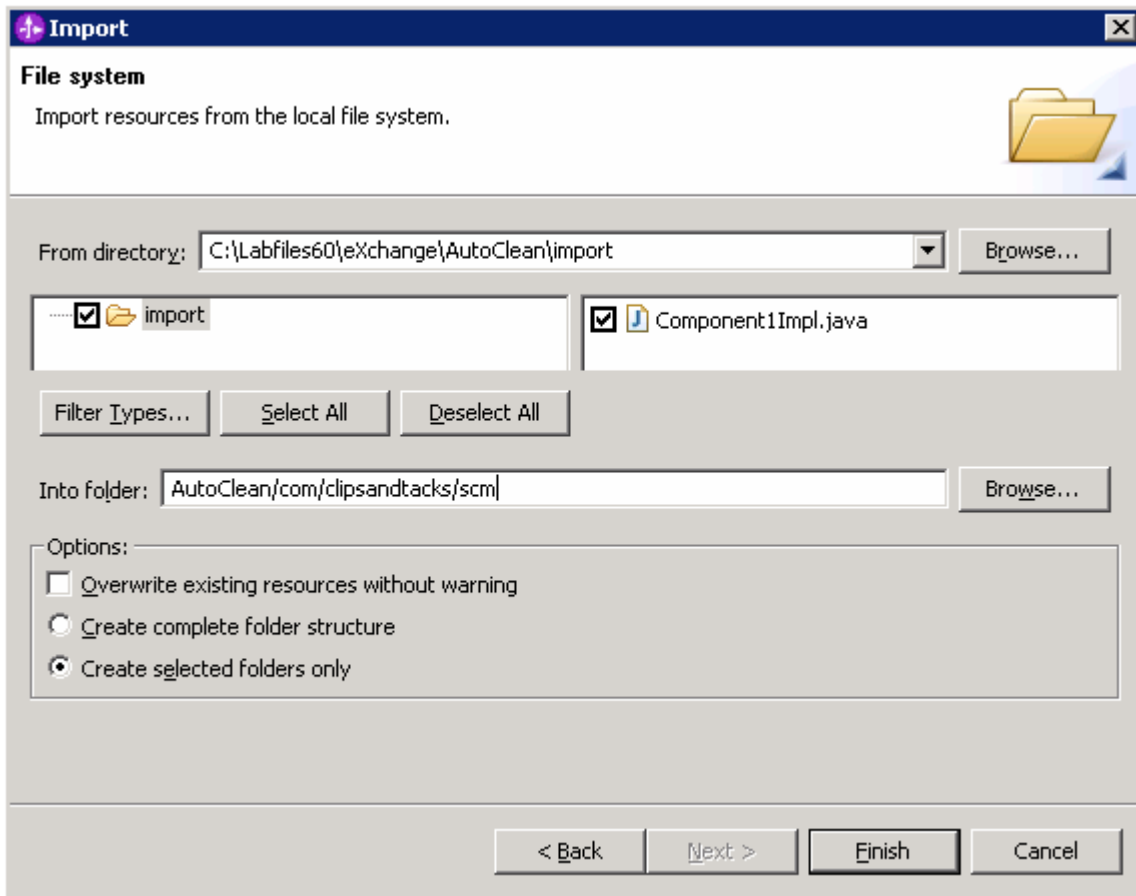
- ___ b. Select **File System**, click **Next**.




___ c. Click **Browse...** and select <LabFiles60>\eXchange\AutoClean\import.



___ d. Click the check box for **Component1Impl.java** and enter **AutoClean/com/clipsandtacks/scm** for the Into Folder value, click **Finish**.

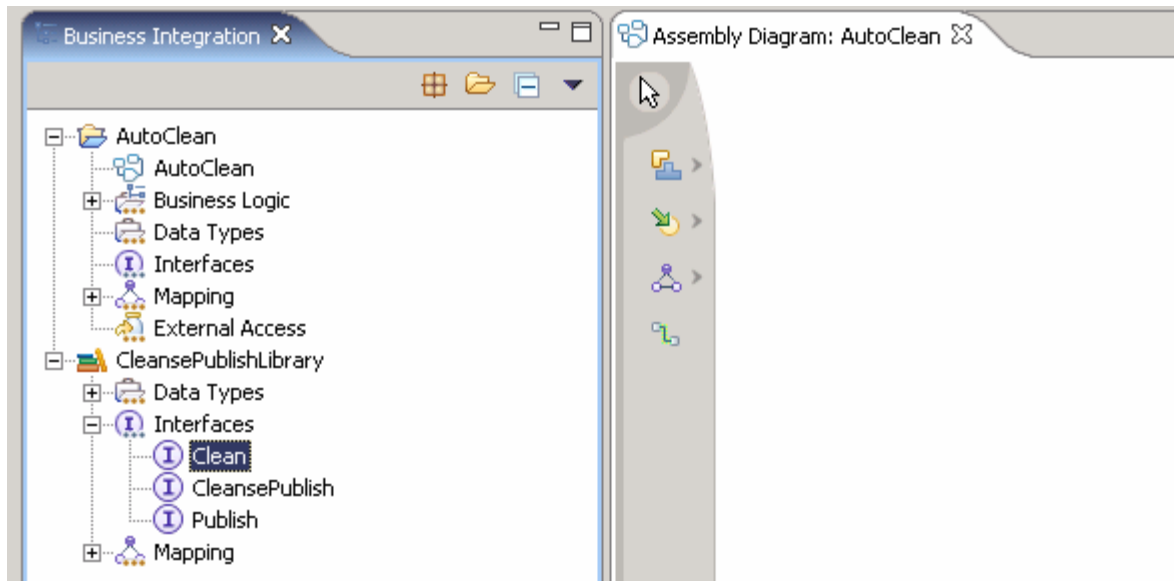


___ 2. Create Java Component.

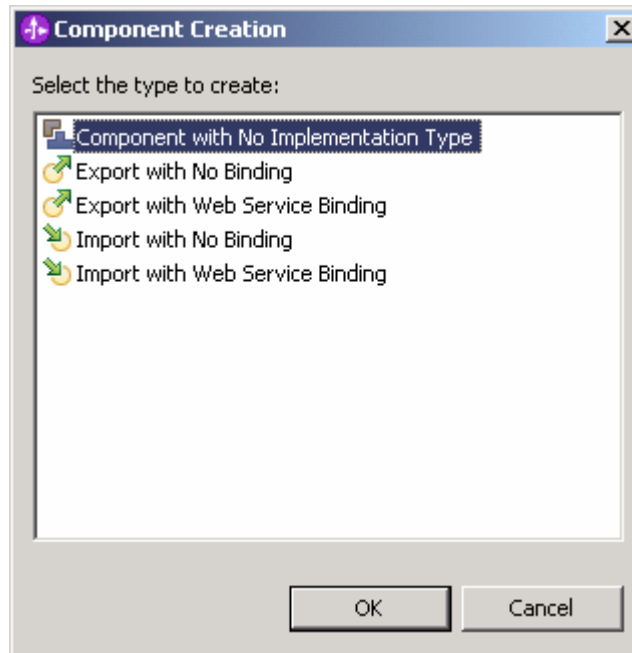
- ___ a. Double click  **AutoClean** folder in Business Integration view to start Assembly Diagram: AutoClean.



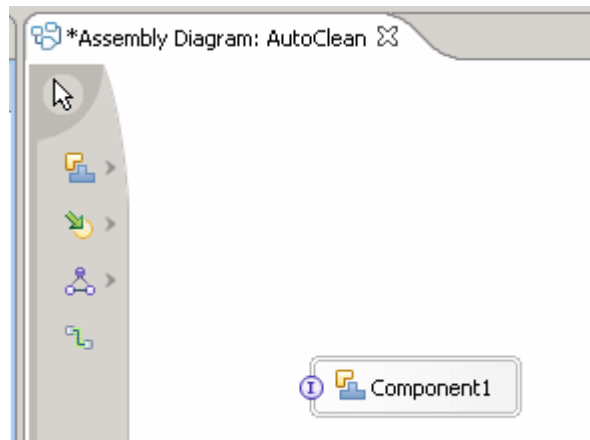
- ___ b. Drag the **Clean** interface from the **CleanePublishLibrary > Interfaces** folder on to the Assembly Diagram.



- ___ c. Accept **Component with No Implementation Type**, click **OK**



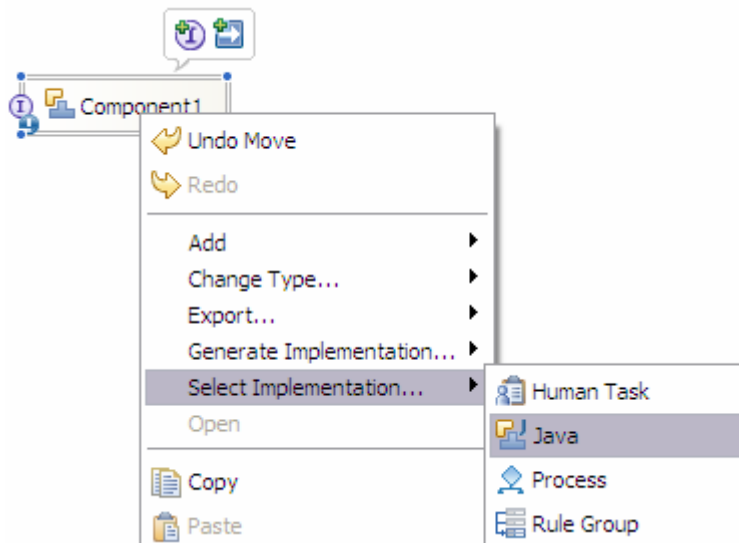
___ d. A new **Component1** will be created.



Note: Optionally, the Component1 display name can be changed to a more descriptive name. Right-click, select show in properties, the display name could be edited. For this lab, it is recommended you leave it as Component1.

___ 3. Pick the Java implementation imported in previous step.

___ a. Right-click on **Component1** in the Assembly Diagram, select **Select Implementation->Java**.



Note: Optionally, a new Java implementation could be generated using the **Generate Implementation...**

- ___ b. Enter **Component1Impl** in the Select entries field, select **Component1Impl** from the Matching types field, click **OK**.



NOTE: What this does is import some java code that uses the Clean interface. Normally you would Generate Implementation and edit that code. However, in this lab there is no need to generate implementation since it has been done for you, and the appropriate code to use in this lab has been added.

- ___ 4. Press **Ctrl+S** to save the workspace. This will also build the workspace.

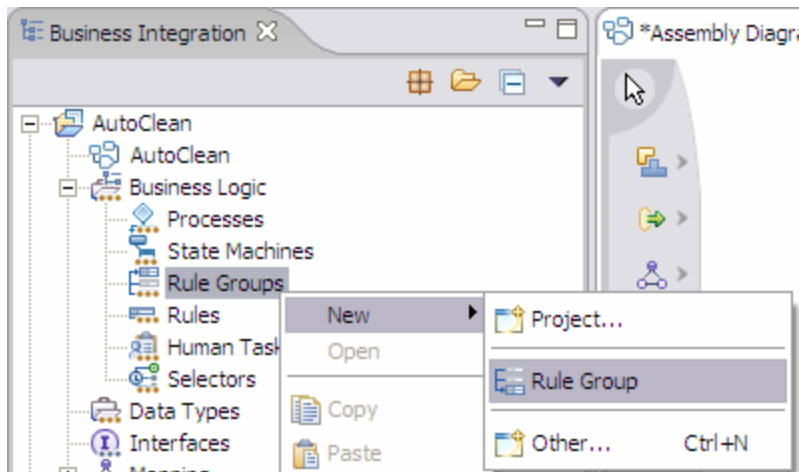
Part 3: Create a Business Rule Group

In this section, you will be creating a Business Rule Group. A Rule Group must be set up before creating a business rule implementation (Rule set or Decision Table). The job of the Rule Group is to group like business rules, and is the SCA component that is exposed as a service (seen by other SCA services). In this example, you will create a Ruleset to see business rule features.

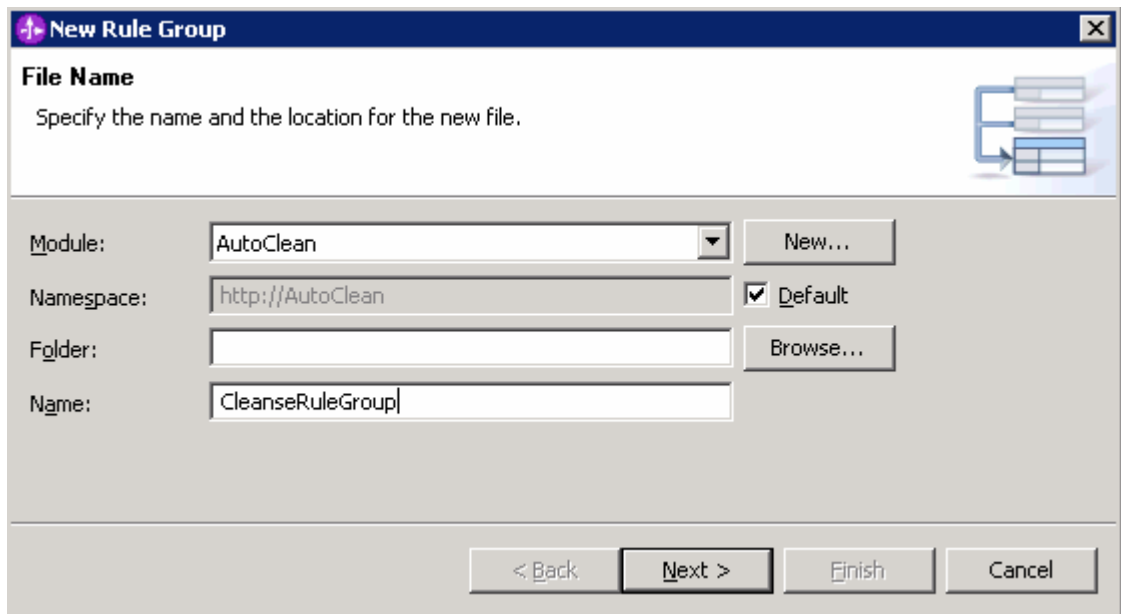
NOTE: You may receive errors in the Problems view as you create components. These errors will come and go as you save/build/finish building pieces.

___ 1. Create a Rule Group.

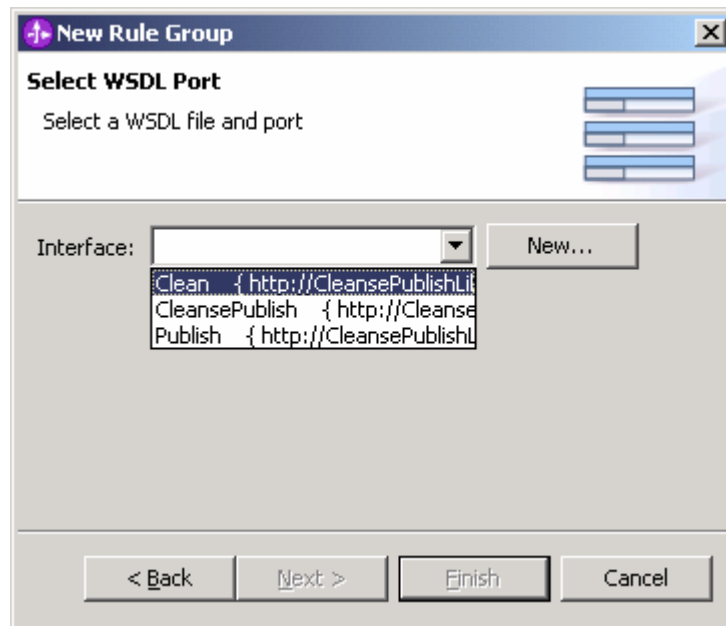
___ a. Right click on the **AutoClean -> Business Logic -> Rule Groups**, select **New->Rule Group**.



___ b. Enter **CleanseRuleGroup** for the Name, click **Next**.



___ c. Select **Clean** for the Interface, click **Finish**.



___ d. Click **cleanseClip** on the Clean Interface to open the Rule Group properties.

Note: An error will display on CleanseClip, but will go away once a selection is entered.

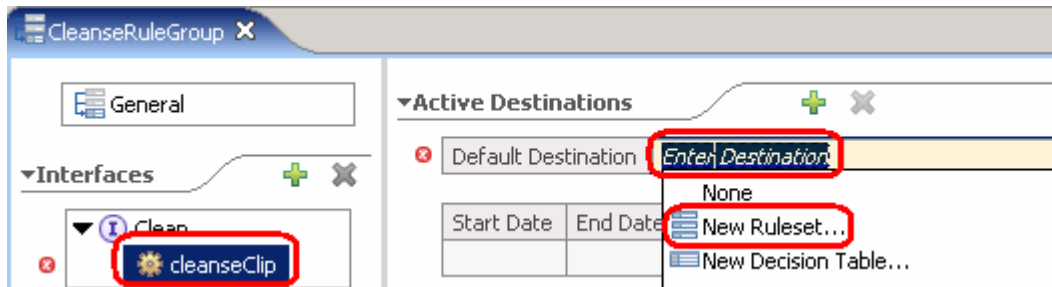
Part 4: Create a Business Rule Set

In this section, you will be creating a Business Rule Set from a Rule Group. The Ruleset is one of the two implementations of a Business Rule, and the easiest one to show how a Business Rule works.

Note: You may receive errors/warnings in the Problems view as you create components. These errors/warnings will come and go as you save/build/finish building pieces.

___ 1. Create a Rule Set.

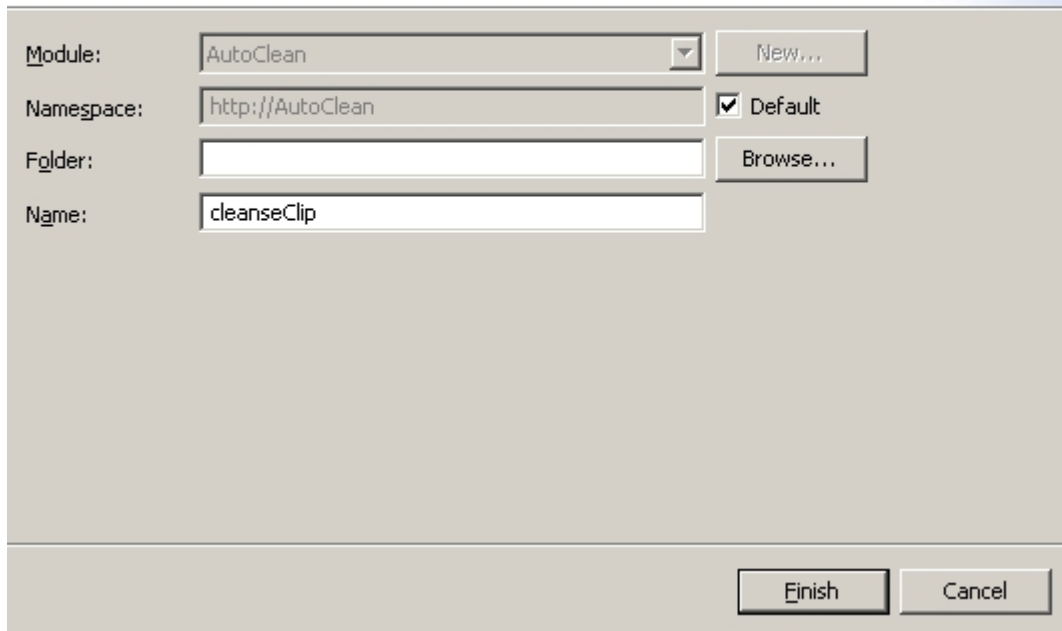
- ___ a. Select the **cleanseClip** under the **Clean** Interface and click on the **Enter Destination** next to Default Destinations, and select **New Ruleset...**





- ___ b. Accept the default values on the New Ruleset screen, click **Finish**.


File Name

Specify the name and the location for the new file.





A screenshot of the 'File Name' dialog box. It has a title bar with 'File Name' and a close button. Below the title bar, there is a text area with the instruction 'Specify the name and the location for the new file.' and a blue icon. The main area contains several fields and buttons: 'Module:' with a dropdown menu set to 'AutoClean' and a 'New...' button; 'Namespace:' with a text field containing 'http://AutoClean' and a checked 'Default' checkbox; 'Folder:' with an empty text field and a 'Browse...' button; 'Name:' with a text field containing 'cleanseClip'. At the bottom, there are 'Finish' and 'Cancel' buttons.

- ___ c. Click  to add a local variable. Enter **sizeRuleExecuted** for the name, **Boolean** for the Type.


▼Variables  	
Name	Type
sizeRuleExecuted	boolean





- ___ d. Click  under Rules to add the first rule, click in the Action area and type **outClipBG = inClipBG**. You will notice that a Content Assist menu will appear when typing. Either use Content Assist or type in manually.
- ___ e. Click **Enter** at the end of the line to enter the initialization of the sizeRuleExecuted variable. The rule should look as follows. If not, simply delete the text of the Action field and repeat selection.

NOTE: To advance to the next entry field in the Action entry, the space bar can be pressed to invoke the next menu.

▼Rules    	
Name	Rule1
Presentation	
Action	outClipBG = inClipBG sizeRuleExecuted = false

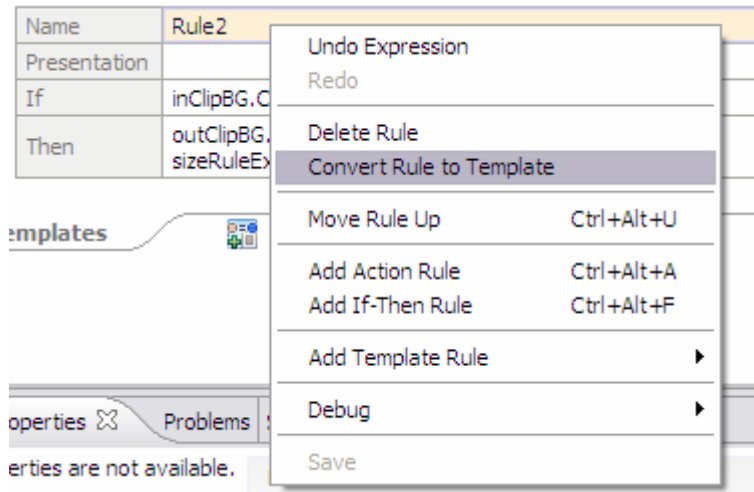
NOTE: There is no condition on Rule1 since it is an action, it will always be run first. The reason outClipBG gets inclipBG is to initialize every element in the Business Graph. The variable **sizeRuleExecuted** is initialized to “false” so that you can test for “true”.

- ___ f. Click  to add an If-Then rule. Click in the If field and select or type **inClipBG.Clip.size**, and **==** from the menu, enter “**small**” for the If field.
- ___ g. Click in the Then section and select or type **outClipBG.Clip.size** and **=** from the selection menu, enter “**small**”, hit enter to advance to the next line, select **sizeRuleExecuted**, and **=**, enter **true**.

▼Rules    	
Name	Rule2
Presentation	
If	inClipBG.Clip.size == "small"
Then	outClipBG.Clip.size = "small" sizeRuleExecuted=true

NOTE: In order to use the Business Rule Editor Web tool to modify the rules of an installed application, the rules must be expressed in the form of a template. This is why you are generating a template of this Business Rule and later in this lab you can see it from the Business Rule Editor Web tool.

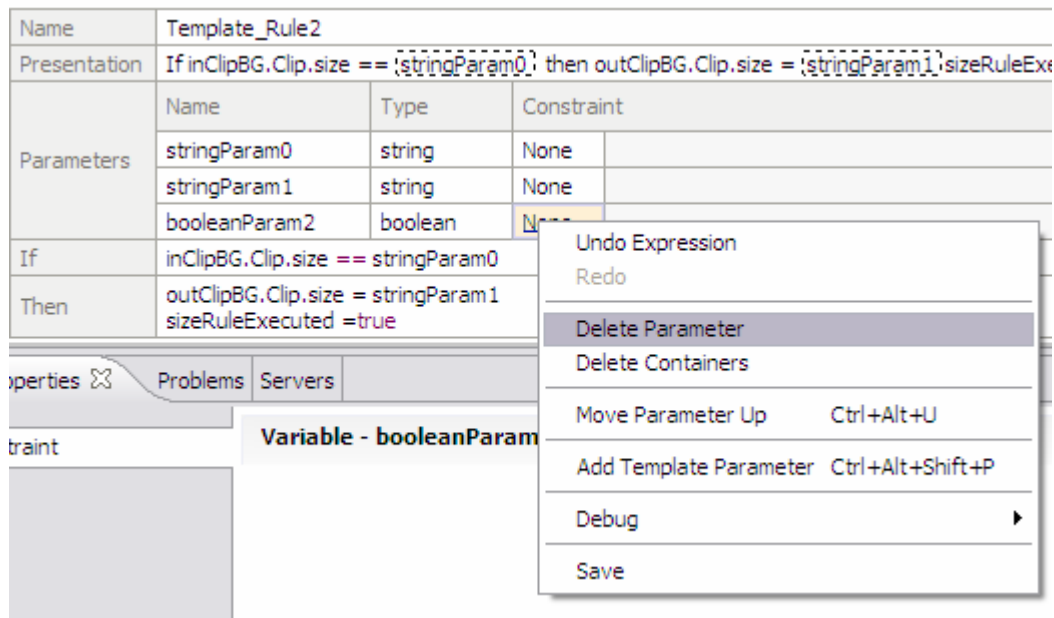
___ h. Right-click on Rule 2, and select **Convert Rule to Template**.



___ i. A template will be generated for Rule 2.

___ j. Change the Template_Rule2 “Then” field: `sizeRuleExecuted = booleanParam2` to `sizeRuleExecuted = true`

___ k. Delete the booleanParam2 from the template since it is always set to true when the rule is executed.



1) The place holder for the Boolean variable {2} should be deleted from the presentation line also (above the Parameters area of template, at end of presentation line). **ALSO** delete the **sizeRuleExecuted=** from the Presentation area.


___ l. The Presentation field can be changed to present the rule in a natural language form. The Presentation field will be displayed in the Business Rules Editor tool. Change the Presentation field in Template_Rule2 to **If the incoming clip size is: {stringParam0}, set the clip size to the standard value: {stringParam1}**.

NOTE: Changing the text in the Presentation field is important because this is all the Business Analyst will see when editing the Business Rules in the Business Rule Editor Web tool.

▼ Templates

Name	Template_Rule2		
Presentation	If the incoming clip size is : {stringParam0} , set the clip size to the standard value : {stringParam1}		
Parameters	Name	Type	Constraint
	stringParam0	string	None
	stringParam1	string	None
If	inClipBG.Clip.size == stringParam0		
Then	outClipBG.Clip.size = stringParam1 sizeRuleExecuted =true		

__ m. Creating a few rules from the template.

- 1) Click , under Rules, and select Template_Rule2.



- 2) The presentation section for Rule3 will look like the following:

Presentation If the incoming clip size is : , set the clip size to the standard value :

Change it to:


Presentation If the incoming clip size is : , set the clip size to the standard value :

- 3) Repeat the above 2 steps to create **3 new rules** from Template_Rule2.





NOTE: **<a blank>** means leave a blank. Delete 'Enter Value' in the output area.

- a) With values **regular** and **<a blank>**.
- b) With values **large** and **large**.
- c) With values **jumbo** and **large**.

__ n. Create a final rule to catch any other types of input.

- 1) Click  to add an If-Then rule.
- 2) Click in the "If" field and select or type **sizeRuleExecuted == false**.
- 3) Click in the Then section and select or type **outClipBG.Clip.size = "?" + inClipBG.Clip.size**. This outputs a question mark before the input text to show that a rule was not fired.

___ o. Here is what the output of the addition of the Rules should look like:

▼ Rules    

Name	Rule1
Presentation	
Action	outClipBG = inClipBG sizRuleExecuted =false

Name	Rule2
Template	Template_Rule2
Presentation	If the incoming clip size : small , set the clip size to the standar value : small




Name	Rule3
Template	Template_Rule2
Presentation	If the incoming clip size : medium , set the clip size to the standar value : □

Name	Rule4
Template	Template_Rule2
Presentation	If the incoming clip size : regular , set the clip size to the standar value : □

Name	Rule5
Template	Template_Rule2
Presentation	If the incoming clip size : large , set the clip size to the standar value : large

Name	Rule6
Template	Template_Rule2
Presentation	If the incoming clip size : jumbo , set the clip size to the standar value : large



Name	Rule7
Presentation	
If	sizRuleExecuted ==false
Then	outClipBG.Clip.size = "?" + inClipBG.Clip.size

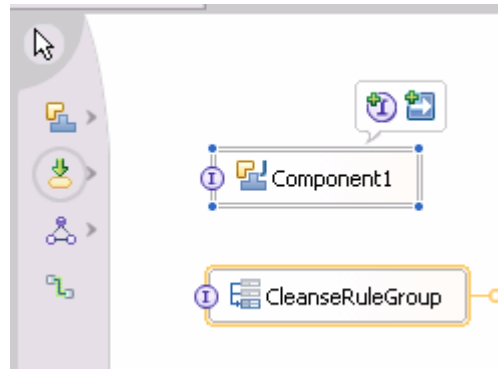
▼ Templates   

Name	Template_Rule2		
Presentation	If the incoming clip size : {stringParam0} , set the clip size to the standar value : {stringParam1}		
Parameters	Name	Type	Constraint
	stringParam0	string	None
	stringParam1	string	None
If	inClipBG.Clip.size == stringParam0		
Then	outClipBG.Clip.size = stringParam1 sizRuleExecuted = true		

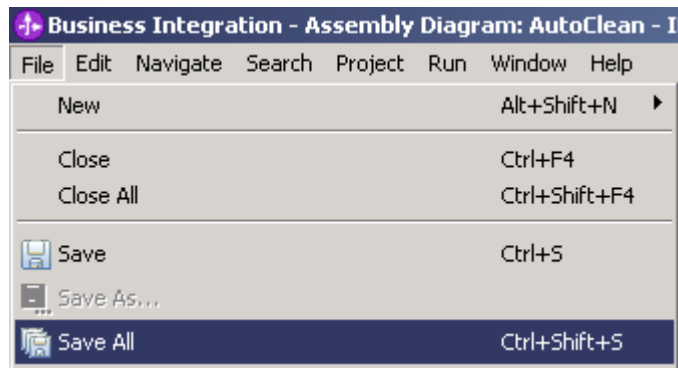
___ p. Save the Rule Set (**Ctrl+S**).

___ q. Add CleanseRuleGroup to the Assembly Editor.

- 1) Go to the **Assembly Editor** by double clicking the  AutoClean folder.
- 2) Drag  CleanseRuleGroup onto the assembly editor.



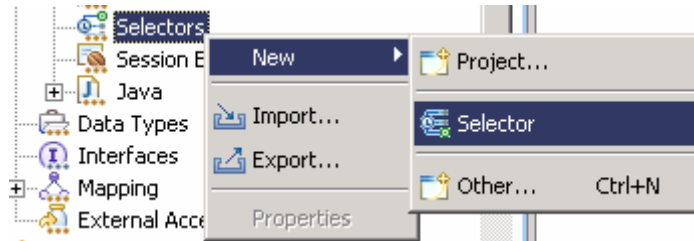
2. Select **File>Save All**, this will build the workspace.



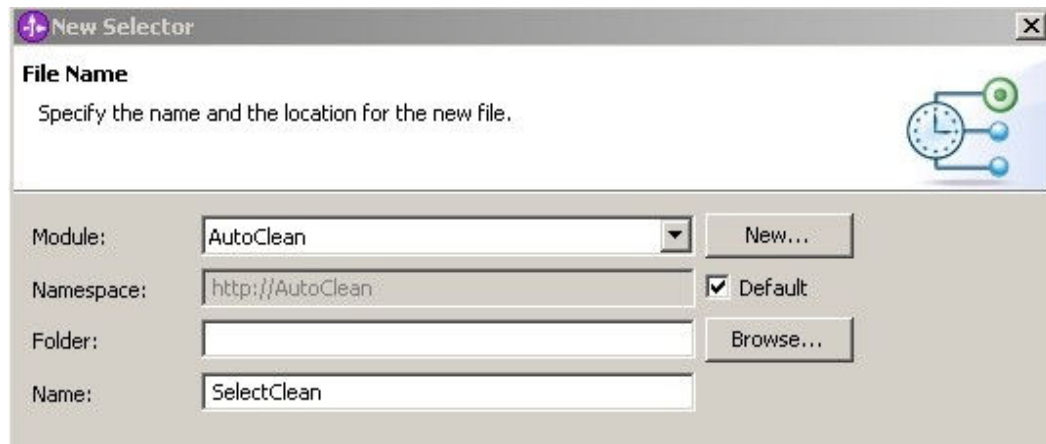
Part 5: Create a Selector

A simple selector will be created to invoke the simple Java business rule, which at some time will switch to use the business rule. You created and imported the component destinations first, so that creating the selector is easier since you already have the destination components in the drop down list.

- ___ 1. Create a new Selector.
 - ___ a. Navigate to **AutoClean > Business Logic > Selectors**.
 - ___ b. Right click on Selectors and select **New >Selector**.

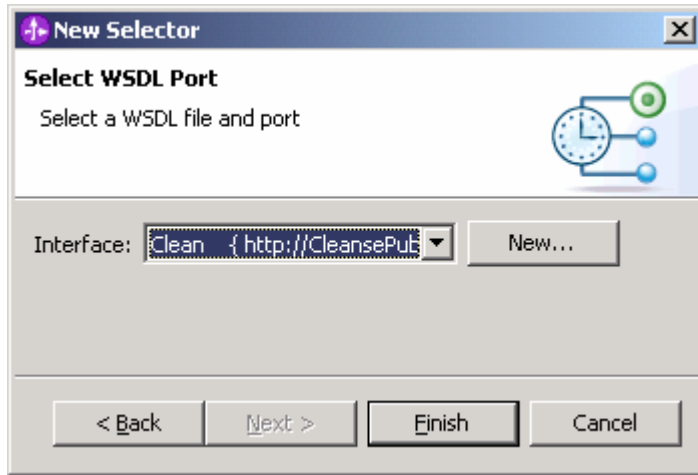


- ___ 2. Enter **SelectClean** for the Name, click **Next**.

A screenshot of the 'New Selector' dialog box. The title bar reads 'New Selector'. Below the title bar, there is a section titled 'File Name' with the instruction 'Specify the name and the location for the new file.' and a small icon of a clock and gears. Below this, there are four input fields and two buttons. The 'Module' field is a dropdown menu with 'AutoClean' selected and a 'New...' button to its right. The 'Namespace' field contains 'http://AutoClean' and has a checked 'Default' checkbox to its right. The 'Folder' field is empty and has a 'Browse...' button to its right. The 'Name' field contains 'SelectClean'.

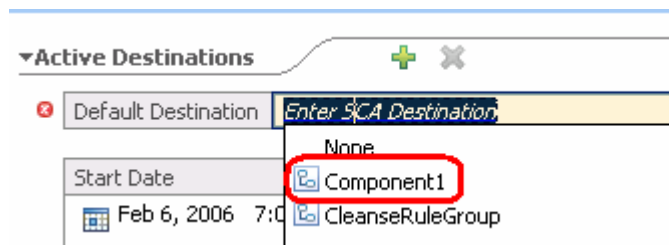
Module:	AutoClean	New...
Namespace:	http://AutoClean	<input checked="" type="checkbox"/> Default
Folder:		Browse...
Name:	SelectClean	

3. Select **Clean** for the interface, click **Finish**.



4. Click on the cleanseClip icon  to open the Selector editor.

5. Click **Enter SCA Destination** next to Default Destination and select **Component1**.

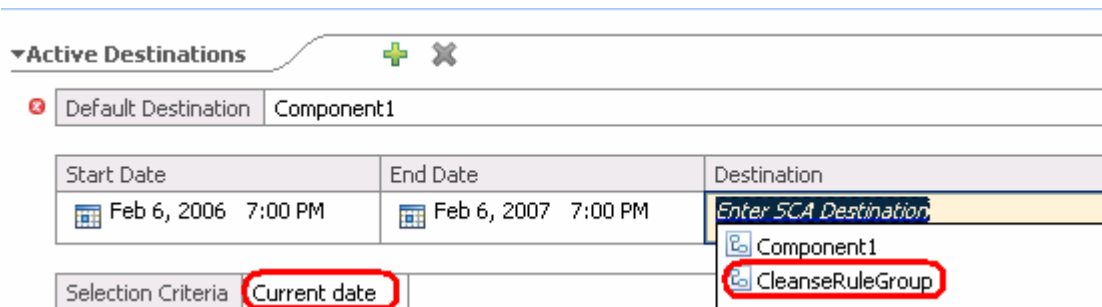


6. Click on the **plus icon** , under Active Destinations, to add date selection entry.

7. Change the time to add **1 hour** to the current time under the **Start Date** field. You will be adding an hour so that in that 1 hour, the selector will switch over from using Component1 (the POJO) to the Business Rule.



8. Click on the **Enter SCA Destination** and select **CleanseRuleGroup**.

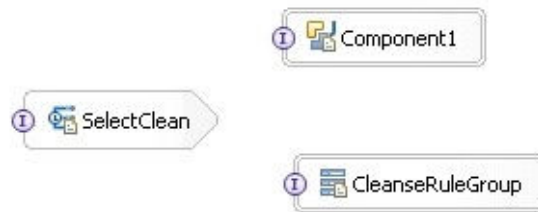
9. Click **Enter Method** for Selection Criteria, select **Current date** (if not already set).



10. Select **File > Save All** to save and build the workspace.

___ 11. Place Selector on Assembly Diagram.

- ___ a. Double Click  AutoClean folder in Business Integration view to open the **Assembly Diagram** for the AutoClean module.
- ___ b. Drag  SelectClean onto the Assembly Editor.
- ___ c. Save your work (**Ctrl+S**).
- ___ d. Your Assembly Editor should have the following:




NOTE: You cannot wire the Selector to the destination components. However, the Selector is already wired to the two target components - Component1 (the POJO) and the CleanseRuleGroup (the Business Rule). The selector is using dynamic wires to point at Component1 and the CleanseRuleGroup under the covers. Meaning the Selector is already “wired” to those destination SCA components. Any editing of the Target Implementations would need the Selector Editor and not the Assembly Editor.

Part 6: Create an Export on Assembly Editor

In order for these components to be accessed from other modules, you must add an Export. The Export acts like a door for other modules to access the modules inside. Export allow others to use this module's services.

___ 1. Open the Assembly Editor.

___ a. Double Click  **AutoClean** folder in Business Integration view to open the Assembly Diagram: AutoClean if not already open. You should have Component1, CleanseRuleGroup, and Selector.

___ 2. Create an Export on the Assembly Editor.

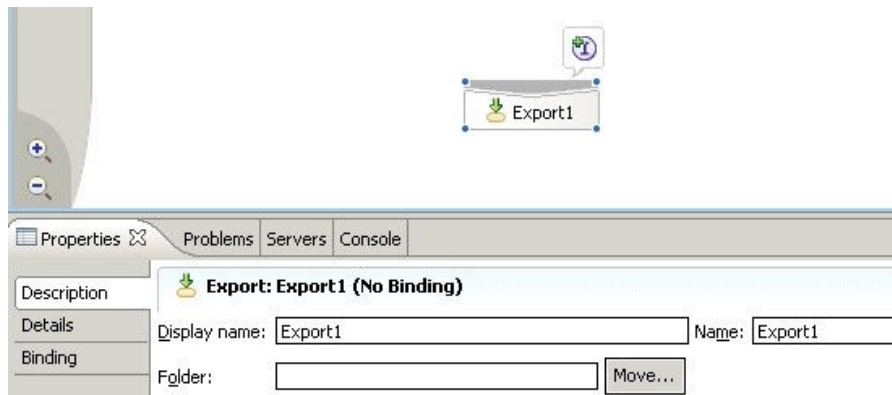
___ a. Open and click the **Export** button on the Assembly Editor.



___ b. After clicking the button, move the cursor over the Assembly Editor and click inside the Assembly Editor empty space to place the Export on the editor. An Export1 icon will appear.

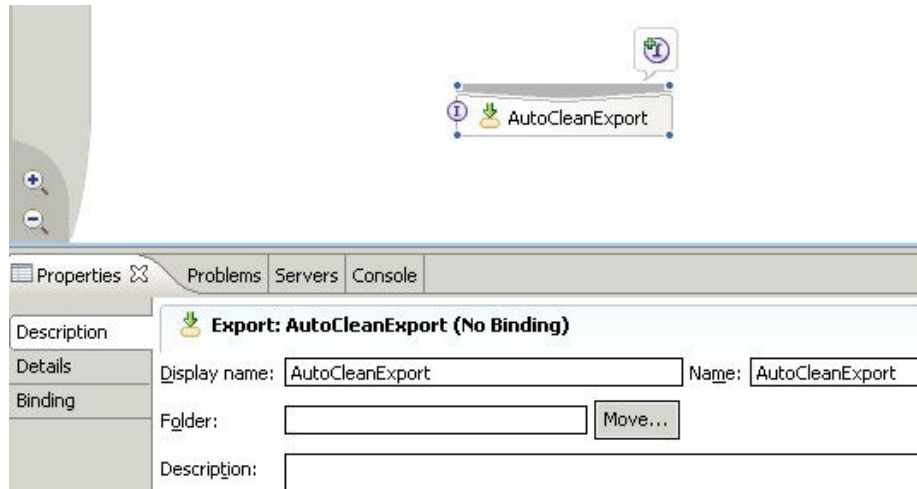
___ 3. Change the **Display Name** and **Name** from **Export1** to **AutoCleanExport**. You are doing this because the AutoClean will be connected with the other modules through this export. Naming this export appropriately allows you to easily know what modules you are connecting to.

___ a. Click on **Export1** in the Assembly Editor so that it is highlighted with focus. (Hint: click on text in icon for one click selecting.)



___ b. Open the **Properties View**.

__ c. Change the **Display Name** and **Name** from **Export1** to **AutoCleanExport**.



___ 4. Add the Clean interface to the AutoCleanExport. The reason you need to add an interface is to define the inputs and outputs of the module, so that requests from clients know what data types the module uses.

__ a. Right click on the **AutoCleanExport** in the Assembly Editor and choose **Add Interface**.



__ b. Select the **Clean** interface when prompted by the Add Interface wizard and click **OK**.

___ 5. Wire the Export to the Selector. Wiring the Export to the Selector creates the bridge that allows other SCA services to use the Selector.

__ a. Select the Wire button  on the Assembly Editor toolbar.

__ b. Click on AutoCleanExport first, and then on the SelectClean selector. You will notice a wire will follow wherever you click the second time. In this case, the first click is on the Export, then on the Selector. Click OK on any window that may pop up.

Your Assembly Editor should now look something like the screen capture below.



____ 6. Select **File > Save All** to save and build the workspace.

You now have created an export so this Selector/Business Rule module can be accessed and used by the other modules.

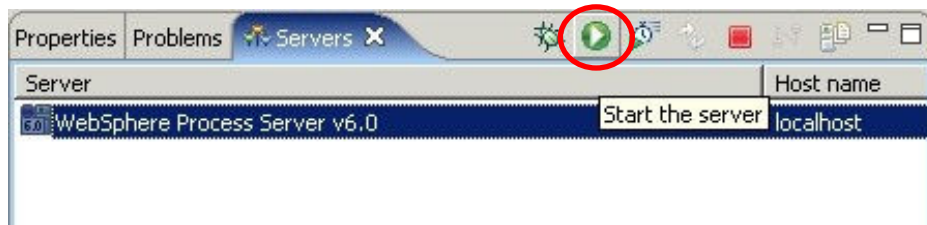
Part 7: Publish the AutoCleanApp

In order to see the Business Rules with the Business Rules Editor Web tool and to test the Business Rules using the Component Test feature in WebSphere Integration Developer, you must publish the application to the WebSphere Process Server. Along with the development tools there is a full version of the WebSphere Process Server that you can use to auto-publish. You could manually install the application using the server Administrative Console. However, this can be accomplished much quicker by adding the project to the server from the development tool.

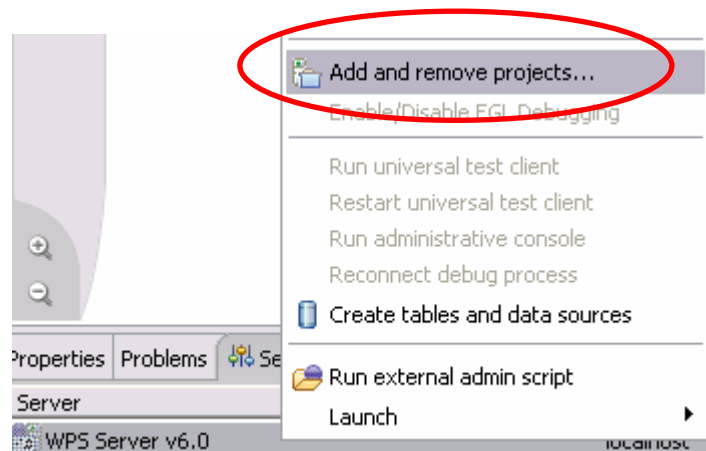
1. Start the server.

If using a remote testing environment, follow the directions provided in [Task: Adding Remote Server to WebSphere Integration Developer Test Environment](#) to add a server to the WebSphere Integration Developer test environment and start it. This is especially true for z/OS, AIX, Solaris remote test environment, where the WebSphere Integration Developer will be remote to the test environment.

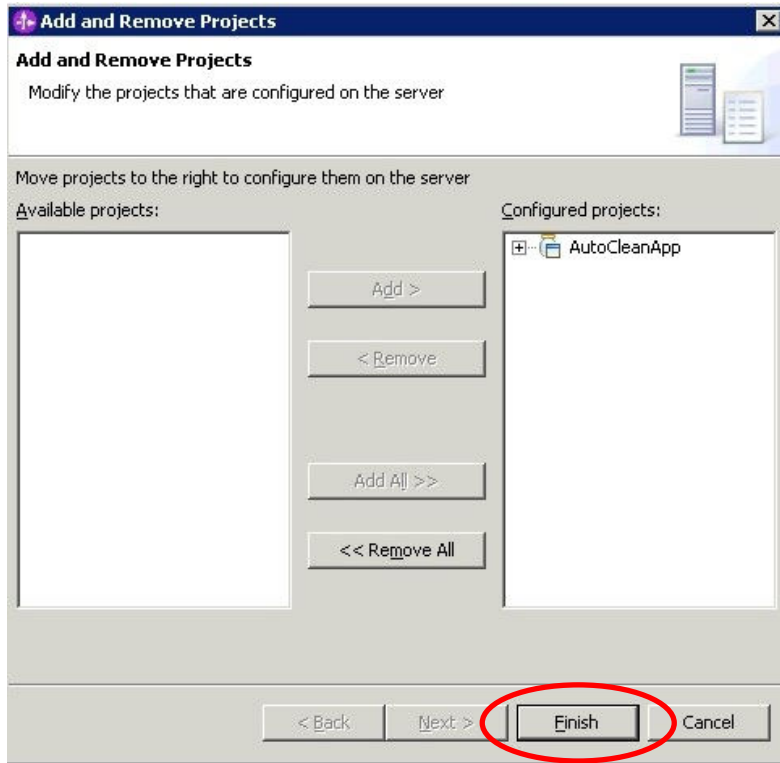
If using a local testing environment, right click on WebSphere Process Server v6.0 from the Server view and wait for it to start successfully.



2. Right click on the WebSphere Process Server in the Servers View and choose **Add and Remove Projects**.



3. Select **AutoCleanApp** and click **Add** and then Click **Finish**.



4. WebSphere Integration Developer will start the server automatically to publish the application if the server is not already started.

Part 8: Update Business Rules using the Business Rule Editor Web tool.

The Business Rule Editor Web tool allows a Business Analyst or a less technical/more business minded person to make business rule changes.

- ___ 1. The Server should already be started and the BRManager Web application already installed.

NOTE: For this lab, we have already installed the BRManager for you. If you are not using a lab system or you are trying this lab on your own system, you can follow the specific steps to install the BRManager Web Tool provided below.

The BRManager Web tool is used to edit, delete and create Business Rules that have been “templated” and are running on the WebSphere Process Server. You can only see Business Rules that have been converted to a template. For instance, Business Rules converted in the tools or WebSphere Integration Developer can be seen in the BRManager Web tool. The Business Rule Editor Web tool allows a Business Analyst or less technical, more business-minded person to make business rule changes without having to restart the server or the application. The Web tool is part of the runtime, and not part of the development tool, WebSphere Integration Developer.

This task assumes that you have WebSphere Integration Developer V6 and the WebSphere Process Server V6 test environment that comes with WebSphere Integration Developer V6 started.

The following files, included in the runtime, are needed to install the Business Rule Editor Web tool:

- 1.) installBRManager.jacl (located at <WAS_HOME>/bin)
- 2.) brmanager.war (located at <WAS_HOME>/installableApps)

These two files are provided for you.

There are **3** requirements to use the jacl file.

- Any security setting of WebSphere Application Server is set to be disabled.
- Code page of command prompt or shell is set to be US environment.
- WebSphere Process Server must be started.

Install instructions

- ___ 2. Run shell (for Linux) or open a command prompt (for Windows) and change directory to <WAS_HOME>/bin (ex: cd C:\WID\runtimes\bi_v6\bin).
- ___ 3. Make sure the code page is set to US by typing **chcp 437** on Win32 machines, **LANG=C** then **LC_CTYPE=C** for Linux machines.

- ___ 4. Run the following command.

Windows: wsadmin.bat -f installBRManager.jacl

Linux: ./wsadmin.sh -f <full_path>/installBRManager.jacl

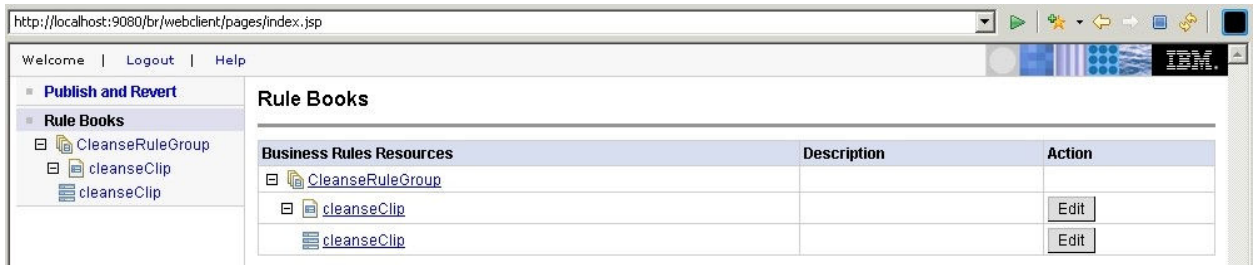
- ___ 5. The business rule Web tool will now be installed and started. You will be able to see the application in the Administrative Console under Enterprise Applications (application is named BRManager in list of apps).

Access the BRManager Web Tool by opening a browser and entering in this URL
<http://localhost:9080/br/webclient/pages/index.jsp>.

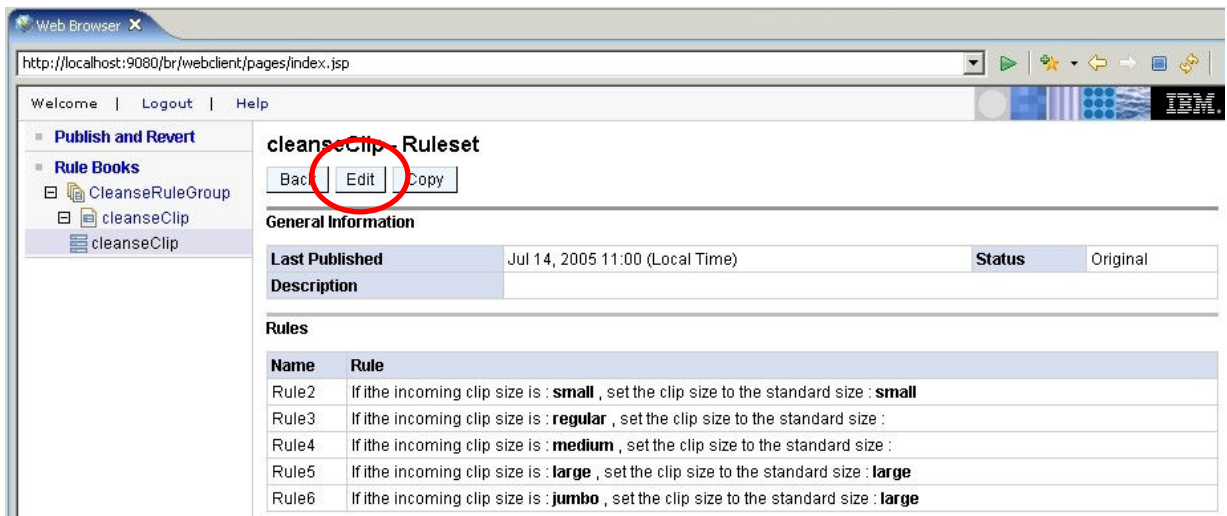


NOTE: If the BRManager does not pick up any business rules at first time, simply restart the server.

6. Once you can access the application, open the CleanseRuleGroup by clicking the **plus box [+]** and open cleanseClip in the same way. Open the cleanseClip ruleset by clicking the **lower cleanseClip link**.

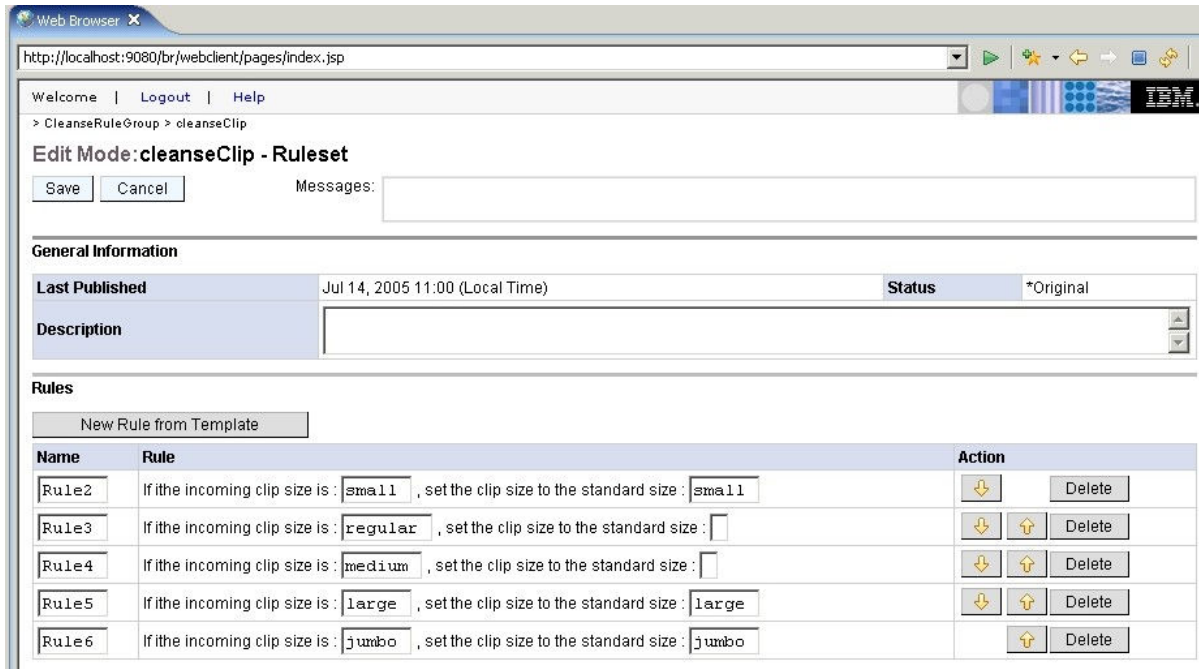


7. These are the rules that were deployed when you published the AutoClean Application. Click the **Edit** button at top to edit the templates you setup.

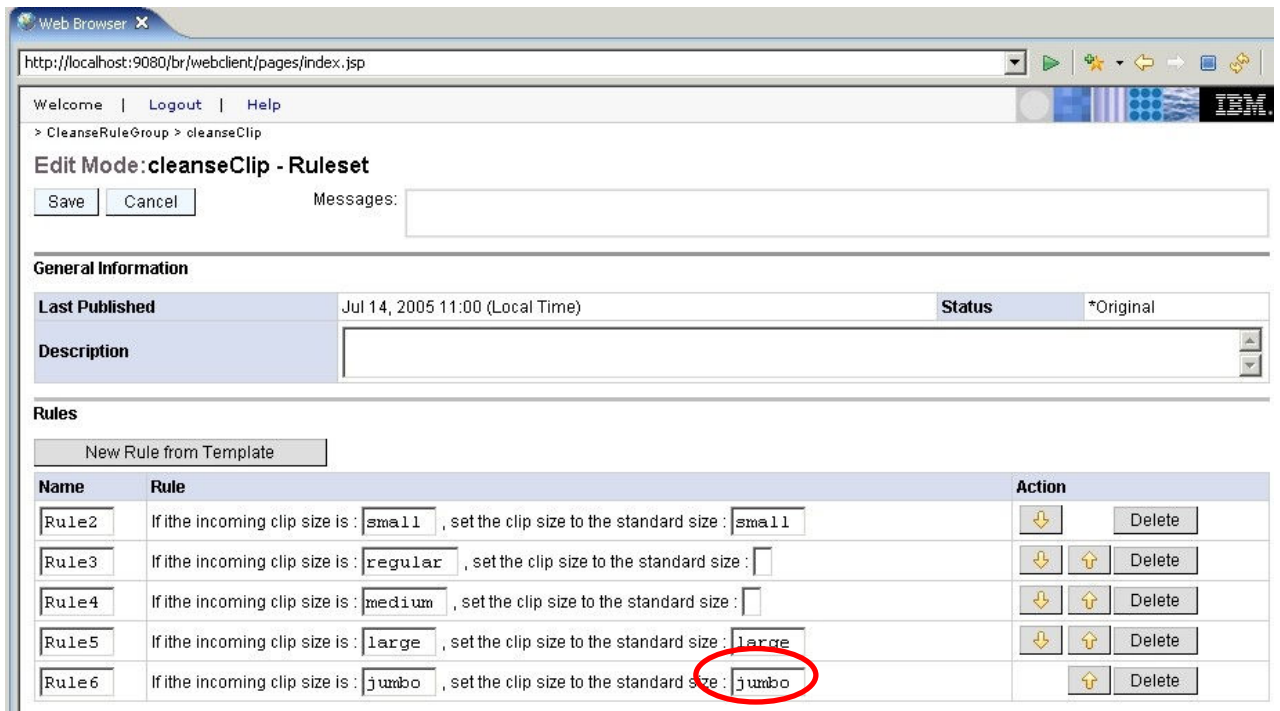


8. With the Web tool, you can change template values, move rules up or down for order of execution (top first, bottom last), and the ability to delete a templated rule. If the rule is not templated, you

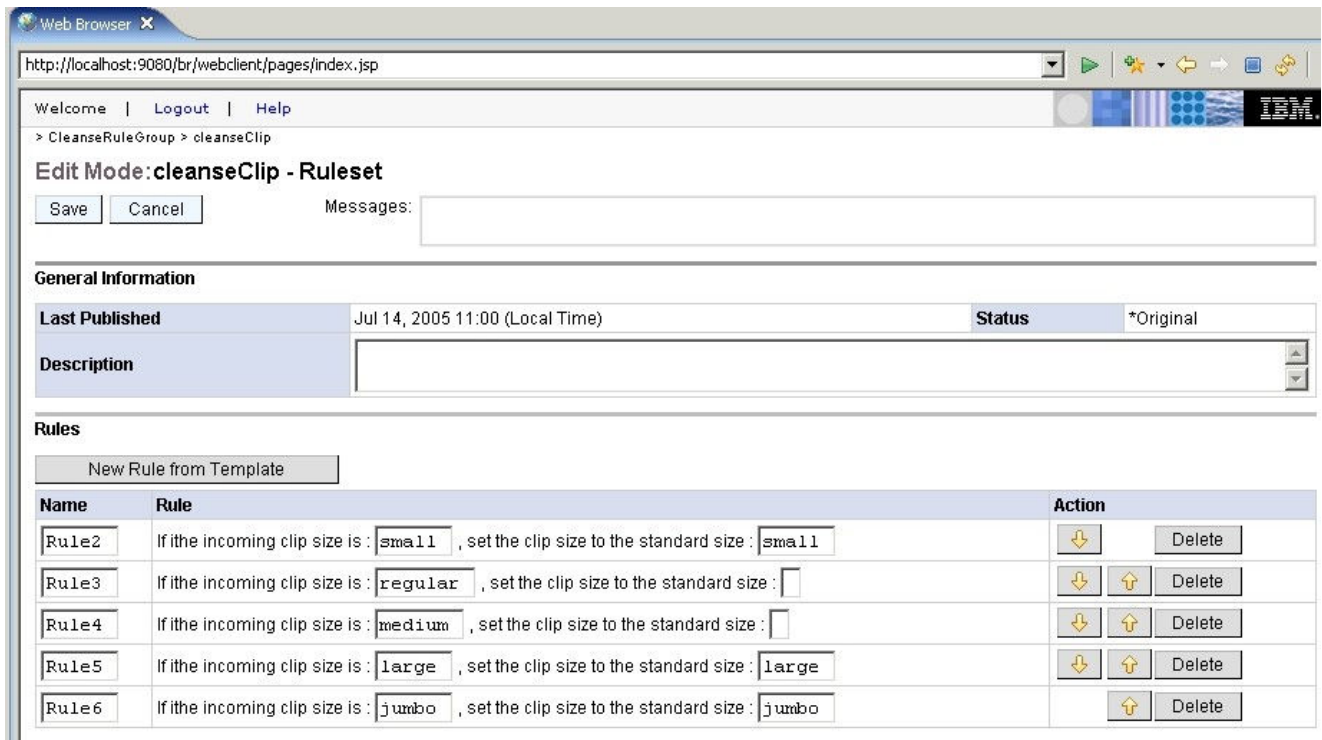
or a business analyst will not be able to see those rules in the Web tool.



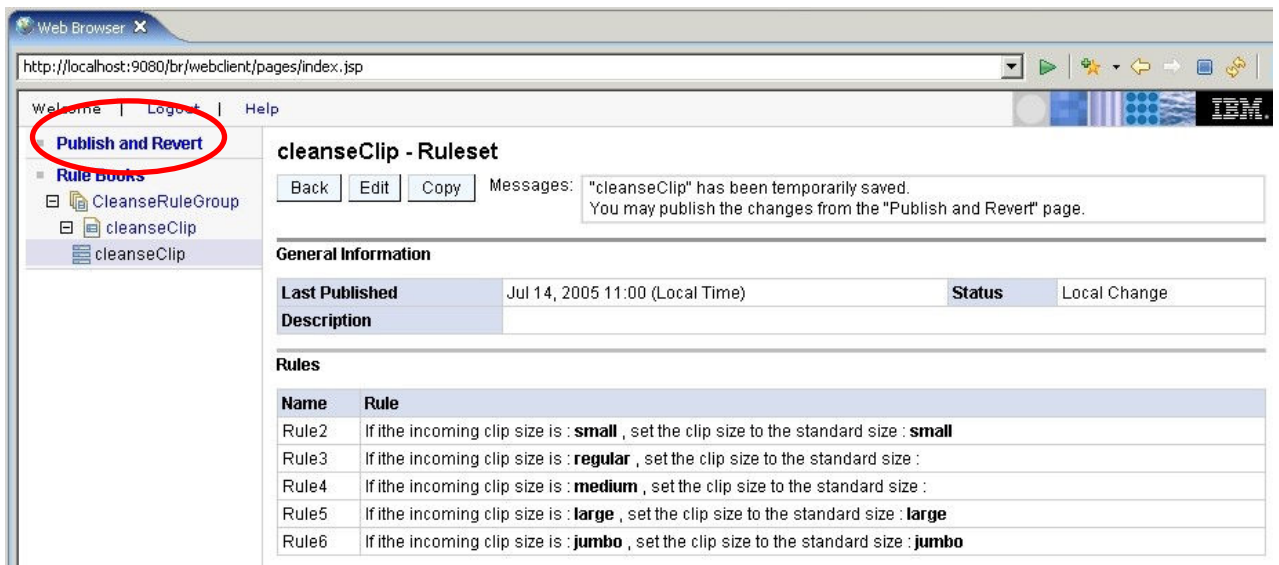
___ 9. Go to the rule with Jumbo and change the result string from **large** to **jumbo**.



10. Click the **Save** button.



11. Click the **Publish and Revert** link to publish the changes to the application server.



12. Now, you can either revert the changes you made or publish them. Click the **Publish** button.



Now, if you were running this application on a production server the input "jumbo" would produce "jumbo", instead of "large" like was initially set up in the business rule.

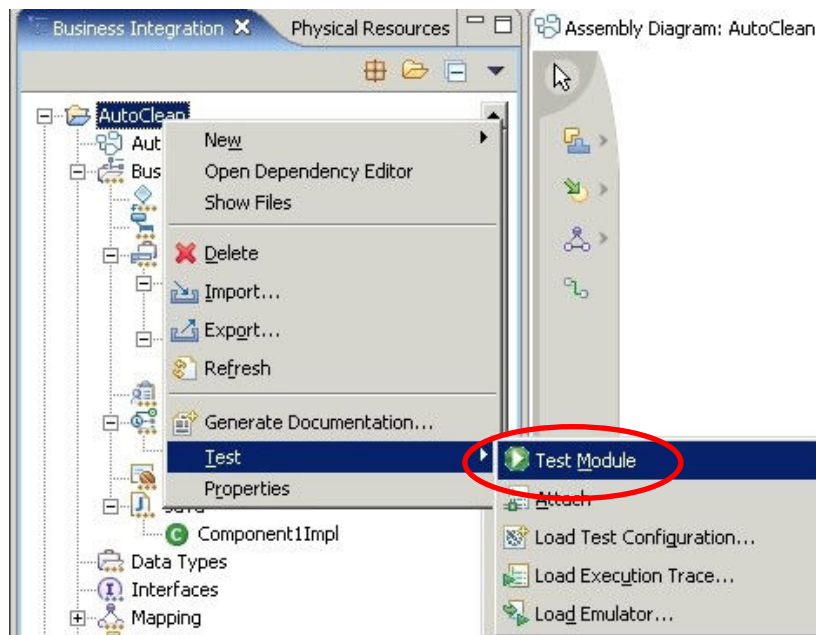
NOTE: If you would like to see how you can edit Selector information, open the Administrative Console by right clicking on WebSphere Process Server V6.0 in the Servers View, and selecting **Run Administrative Console**. Log in using any username and navigate to Application servers > server1 > Selectors and take a look at the options.

Part 9: Test the Business Rule using Component Test

After wiring the components together, its time to test and see if the business rule works. In this section, you will learn to test a business integration module using the component test feature of WebSphere Integration Developer V6.0.1.

Component test is a feature of the Eclipse-based WebSphere Rational® products from which WebSphere Integration Developer is built upon.

- ___ 1. Make sure the server is started. Component Test will not run if not started.
- ___ 2. Run Component Test on the cleanseClip Business Rule.
 - ___ a. Right click on the module (in this case the AutoClean module, the top most folder) and select **Test > Test Module**.



NOTE: You can also open the AutoClean Assembly Editor and right click on the component you would like to test and select Test Component.

- ___ b. Look under detailed properties section of the Component Test.
 - 1) Under configuration, you should have **Default Module Test**. For Module, you should have **AutoClean**.
 - 2) Under Component you should have **CleansRuleGroup**. If you do not, select the drop down menu and choose **CleanseRuleGroup**. CleanseRuleGroup is the Rule Group that points to cleanseClip rule set.
 - 3) The interface you will be using is the **Clean** interface, and the operation you are testing is **cleanseClip**.

4) For the initial request parameters, change the **inClipBG.Clip.size** to **small**. This will send the Clip size of “small” through the flow and should output “small”.

___ c. After the detail properties are set, click the **Continue** button just below the Detail Properties.

▼ Detailed Properties

Configuration: Default Module Test

Module: AutoClean

Component: **CleanseRuleGroup**

Interface: Clean

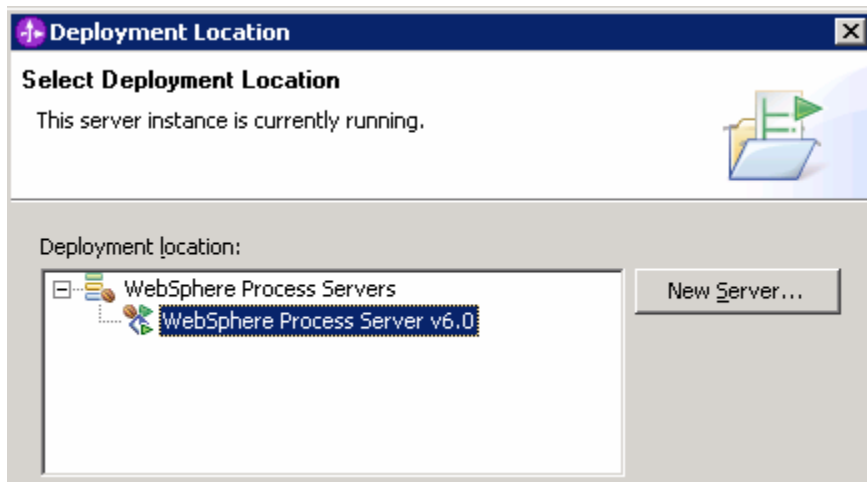
Operation: cleanseClip

Initial request parameters

Name	Type	Value
[-] inClipBG	ClipBG	
verb	string	<null>
[-] Clip	Clip	
clipID	string	
GLN	string	
clip	string	
size	string	small
color	string	
brand	string	
retailItems	ClipItem []	<null>

Data Pool Continue

___ d. Expand **WBI 6.0 Server** in the Deployment Location window and select the appropriate WebSphere Process Server on which you want to deploy the application. This is especially true for z/OS, AIX, Solaris remote test environment, where the WebSphere Integration Developer will be remote to the test environment. Click Finish



___ e. Your return parameter should be **small**. If you have a ? before small, that means the Business Rule did not fire. If you did not get a return, then Component Test did not run correctly, run again.

- ___ f. Run the test again with **jumbo** and **regular** to check correct returns.

Homework question: Why is it when you test with jumbo as the input, you get jumbo as the output (large is still the response for jumbo in the tool)?

- ___ 3. Clean up test environment.

- ___ a. Right-click on appropriate WebSphere Process Server on which you deployed the applications in the Servers view and select Add and remove projects...
- ___ b. Click << Remove All.
- ___ c. Click Finish.

- ___ 4. Stop the Server

- ___ a. If using the local WebSphere Process Server, Right click on WebSphere Process Server v6.0 server from the Servers view and select Stop from the context menu.
- ___ b. If using a remote testing environment, follow the directions provided in [Task: Adding Remote Server to WebSphere Integration Developer Test Environment](#) to stop a server.

What you did in this exercise

In this lab you imported a Java implementation for running a selector/business rule scenario. You then created a rule group in order to create a business rule (rule set). Once the business rule implementation was set, you created a selector and finished wiring the example. When all was wired, you tested using the component test feature within WebSphere Integration Developer. To change the business rule templates, you installed and used the Business Rule Editor application.

Solution Instructions

- ___ 1. Follow the directions in the task [Initialize the Workspace for a Lab Exercise](#), using the following values:

<WORKSPACE>

C:\Labfiles60\exchange\AutoClean\workspace

<PROJECT_INTERCHANGE>

C:\Labfiles60\exchange\AutoClean\solution\AutoClean_P1.zip

<MODULE>

n/a

<DEPENDENT_LIBRARIES>

n/a

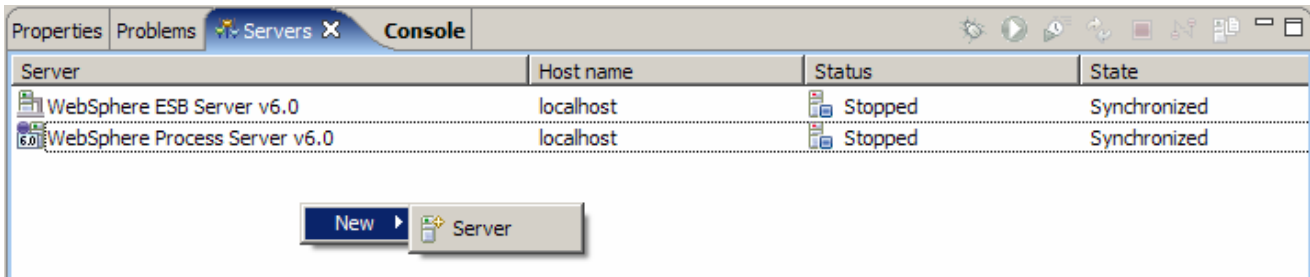
- ___ 2. Continue with **Part 7: Publish the AutoCleanApp**.

Task: Adding Remote Server to WebSphere Integration Developer Test Environment

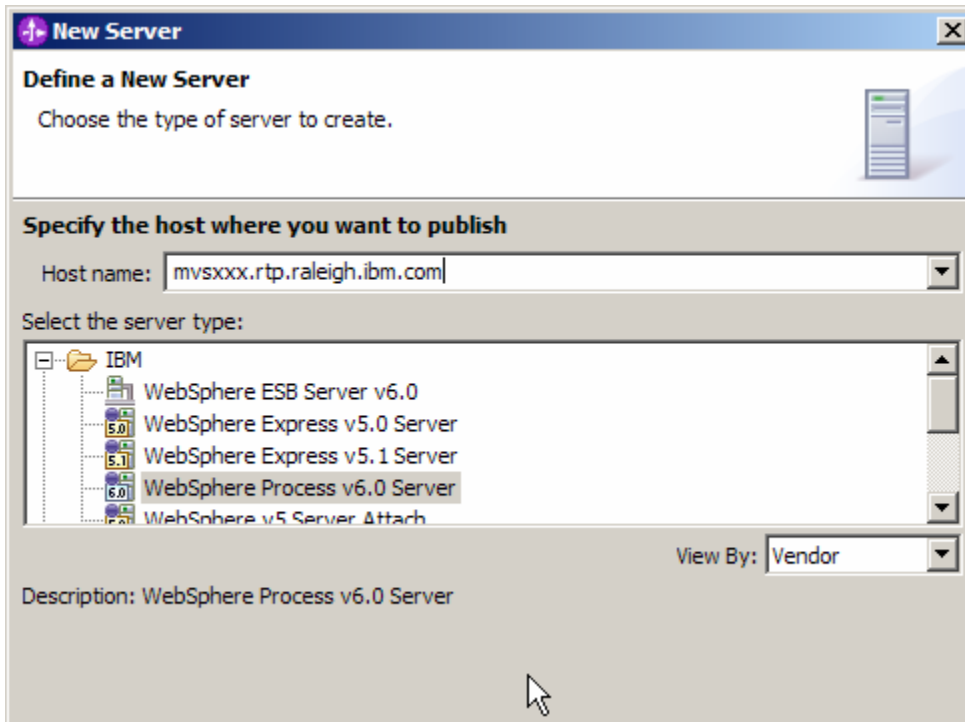
This task describes how to add a remote server to the WebSphere Integration Developer Test environment. The sample will use a z/OS machine.

Create a new remote server.

1. Right click on the background of the Servers view to access the pop-up menu.
2. Select **New > Server**.



3. Specify hostname to the remote server, **<HOSTNAME>**.
4. Ensure that **'WebSphere Process v6.0 Server'** is highlighted in the server type list.



5. Click **Next**.
6. On the WebSphere Server Settings page, select the radio button for **RMI** and change the ORB bootstrap port to the correct setting (**<BOOTSTRAP_PORT>**).

New Server

WebSphere Server Settings

Input settings for the new WebSphere server

WebSphere profile name: []

Server connection type and admin port

RMI (Better performance)

ORB bootstrap port: [9131]

SOAP (More firewall compatible)

SOAP connector port: [8880]

Run server with resources within the workspace

Security is enabled on this server

Current active authentication settings:

User ID: []

Password: []

Server name: [server 1]

Server type

BASE, Express or unmanaged Network Deployment server

Network Deployment server

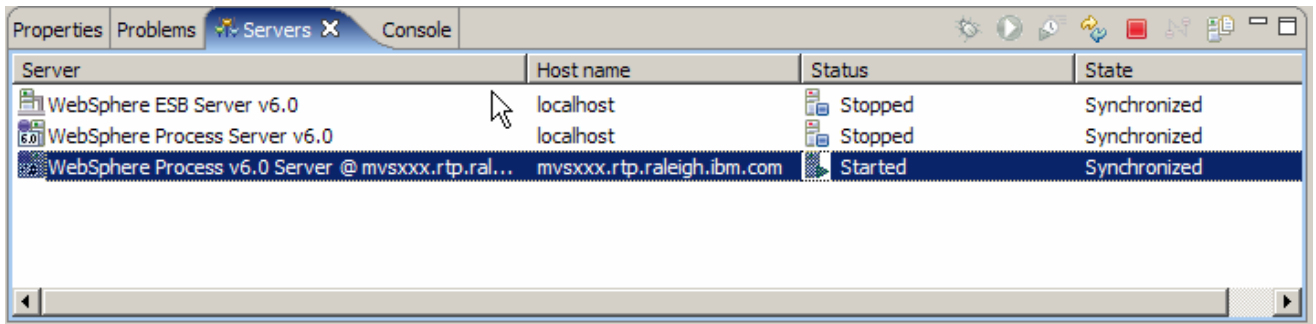
Network Deployment server name: []

The server name is in the form of:
<cell name>/<node name>/<server name>
For example, localhost/localhost/server1.

Click this button to detect the server type.

< Back Next > Finish Cancel

7. Click **Finish**.
8. The new server should be seen in the Server view.



Start the remote server if it is not already started. WID does not support starting remote servers from the Server View.

9. From a command prompt, telnet to the remote system if needed:

```
'telnet <HOSTNAME> <TELNET_PORT>'
```

```
userid : <USERID>
```

```
pw : <PASSWORD>
```

10. Navigate to the bin directory for the profile being used:

```
cd <WAS_HOME>/profiles/<PROFILE_NAME>/bin
```

11. Run the command file to start the server: `./startServer.sh <SERVER_NAME>`

12. Wait for status message indicating server has started:

```
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server c11sr01 open for e-business; process id is 0000012000000002
```