© Copyright IBM Corporation 2005 All rights reserved

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 4
Part 2: Import Simple Java Implementation (a POJO) 9
Part 3: Create a Business Rule Group 16
Part 4: Create a Business Rule Set 18
Part 5: Create a Selector 24
Part 6: Create an Export on Assembly Editor 27
Part 7: Publish the AutoCleanApp 30
Part 8: Update Business Rules using the Business Rule Editor web tool 31
Part 9: Test the Business Rule using Component Test 37
What you did in this exercise 39
Solution Instructions 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 installed
- WebSphere Process Server V6 test environment installed
- Sample code in the directory C:\Labfiles60 (Windows) or /tmp/LabFiles60 (Linux)

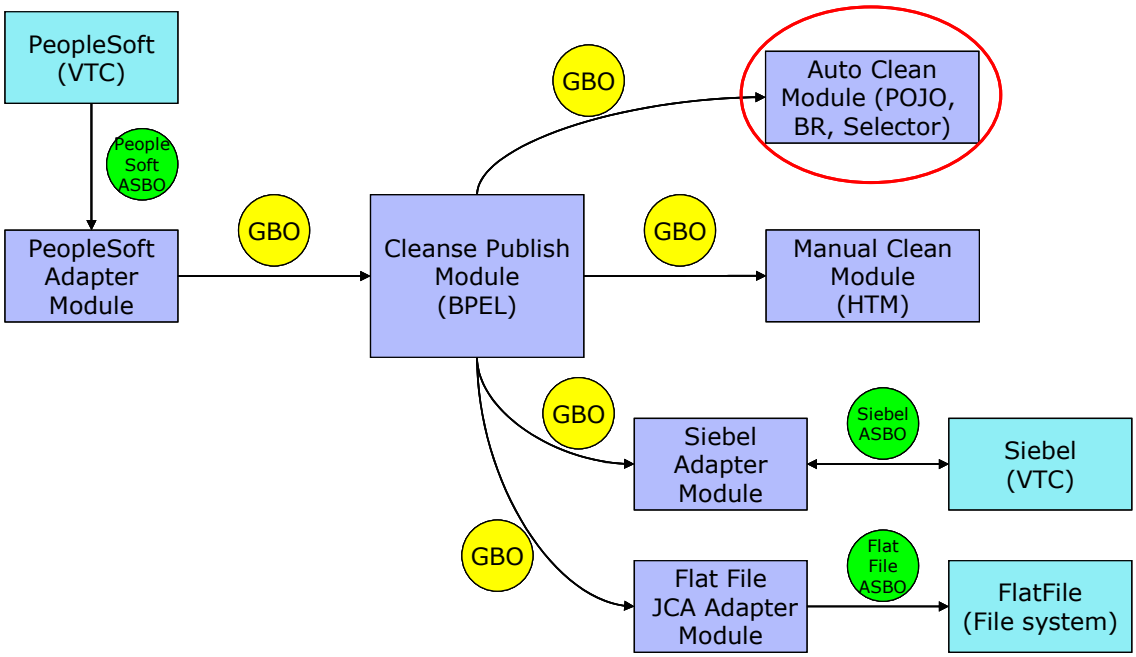
© Copyright IBM Corporation 2005. All rights reserved

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

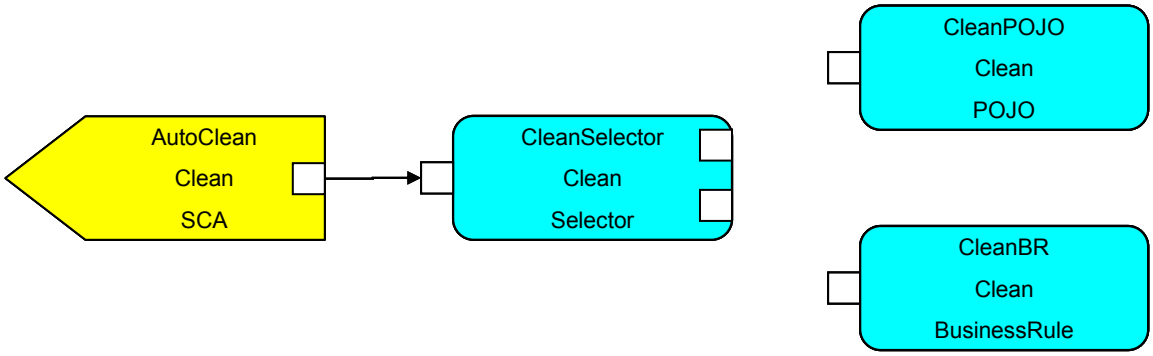
Introduction

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 is choice is determined by a date, either the current date or a date variable mined using Java or Xpath parameters.



© Copyright IBM Corporation 2005. All rights reserved

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 execute 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	
<WPS_HOME>	<WID_HOME>\runtimes\bi_v6	
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles60\AutoClean\Solution	/tmp/AutoClean/Solution
<WORKSPACE>	C:\Labfiles60\eXchange\ AutoClean\workspace	/tmp/Labfiles60/eXchange/AutoClean/ workspace

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/

© Copyright IBM Corporation 2005. All rights reserved

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_PI.zip

<MODULE>

AutoClean

<DEPENDENT_LIBRARIES>

CleansePublishLibrary

Task: Initialize the Workspace for a Lab Exercise

This task is a common description for how to initialize your workspace for doing any one of the eXchange labs. The individual labs each specify variable values for you to substitute for the variables in this description. The variables are:

<WORKSPACE> - the directory path to the workspace.

<PROJECT_INTERCHANGE> - a project interchange file to import into your workspace (if required)

<MODULE> - the name of a new business integration module to be created (if required).

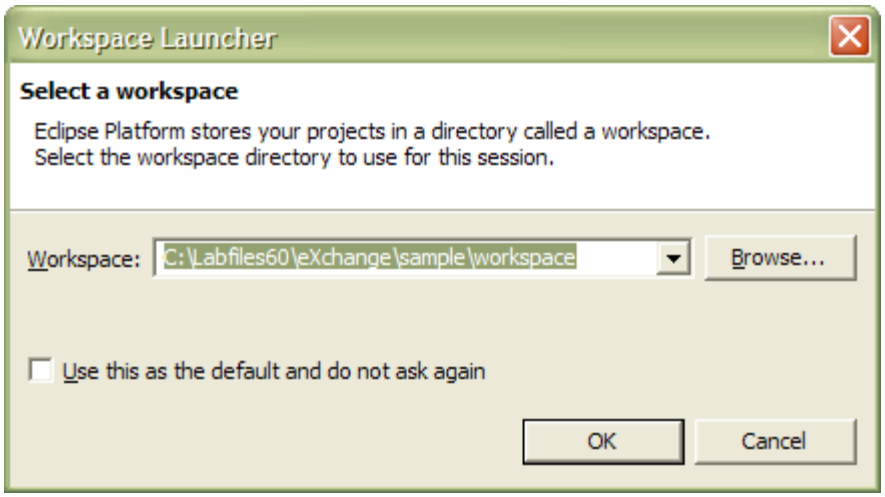
<DEPENDENT_LIBRARIES> - any libraries on which the module is dependent (if required).

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

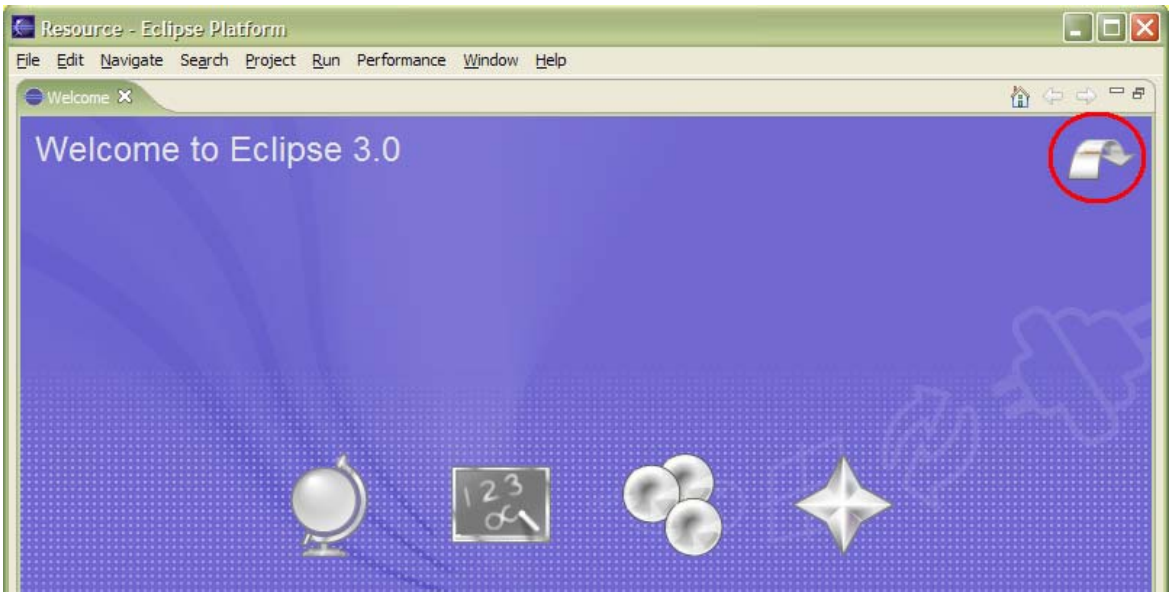
__ a. From Windows Explorer, navigate to the **<WID_HOME>** directory and double click on wid.exe

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

© Copyright IBM Corporation 2005. All rights reserved



__ c. When WebSphere Integration Developer V6.0 opens, close the **Welcome page** by clicking on the Go to the workbench icon (bent over arrow at top-right)



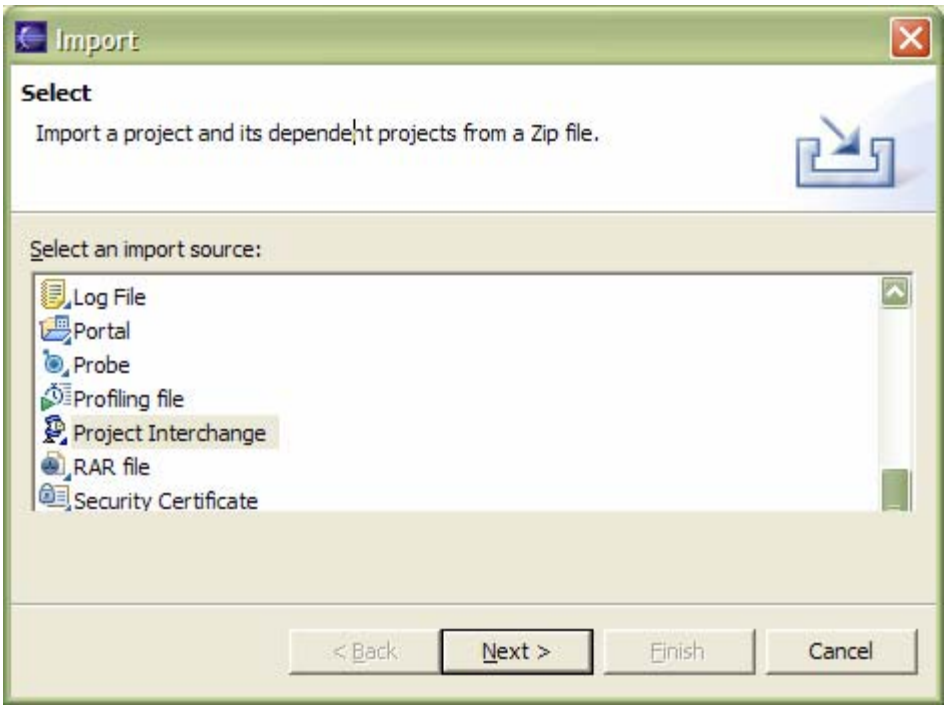
__ d. Ensure you are in the **Business Integration** perspective (See the task **Switching Perspectives within a workspace**)

____ 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**

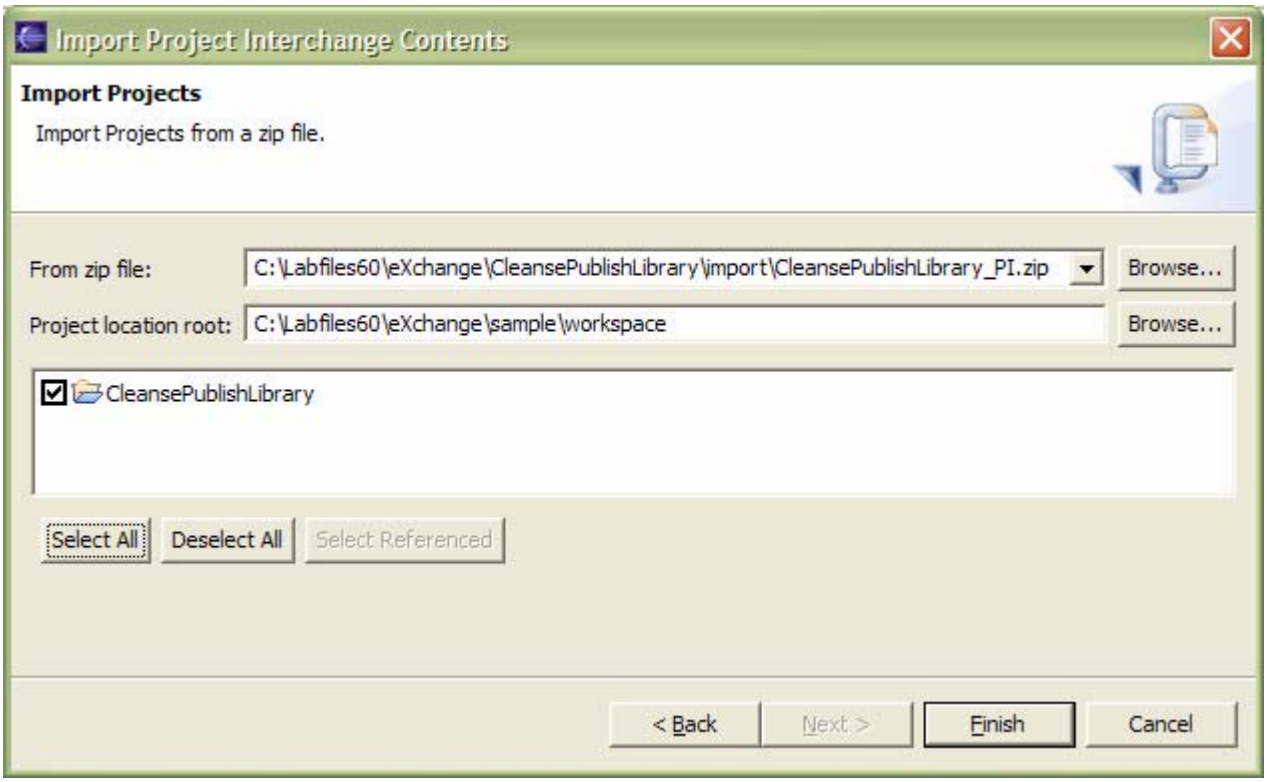
© Copyright IBM Corporation 2005. All rights reserved



__ c. Click **Next**

__ d. In the Import Projects dialog, initialize the From zip file: field to **<PROJECT_INTERCHANGE>**

__ e. Click the **Select All** button



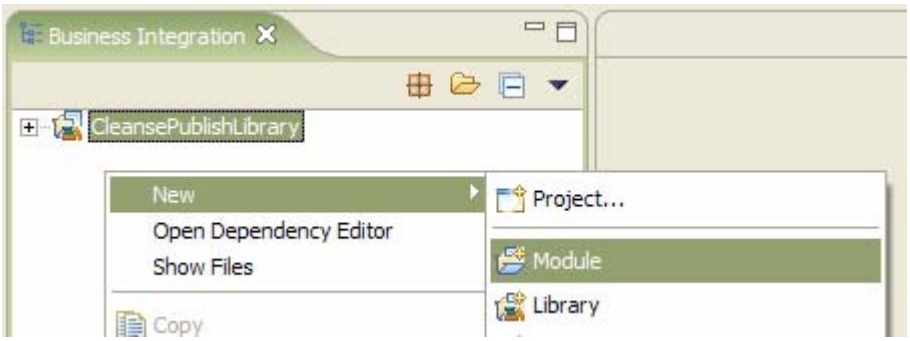
__ f. Click **Finish**

© Copyright IBM Corporation 2005. All rights reserved

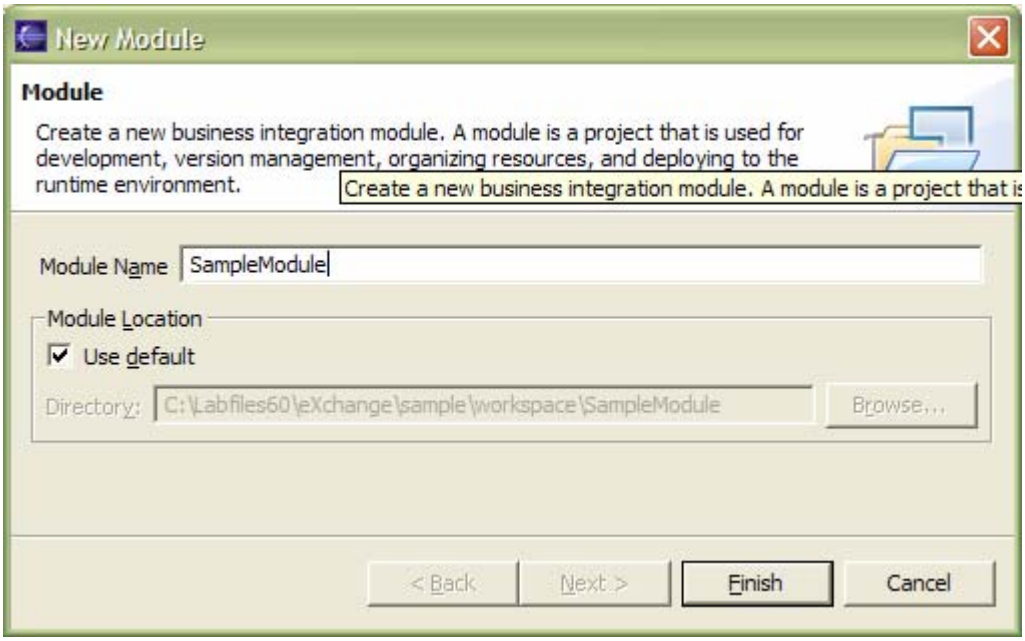
____ 4. If this lab requires that you create a Business Integration module called **<MODULE>**, complete these steps

__ 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



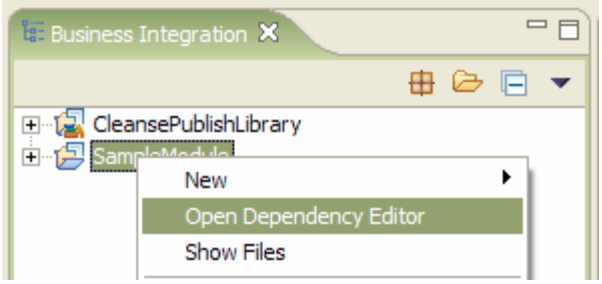
__ d. Click **Finish**

____ 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**

© Copyright IBM Corporation 2005. All rights reserved



__ 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

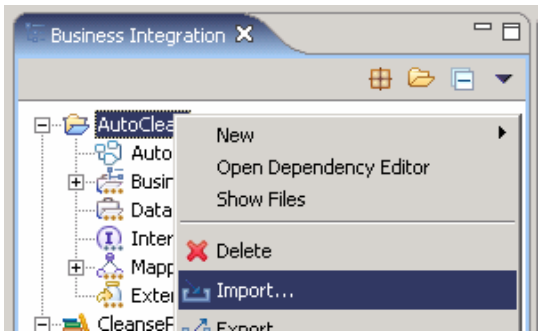
© Copyright IBM Corporation 2005. All rights reserved

Part 2: Import Simple Java Implementation (a POJO)

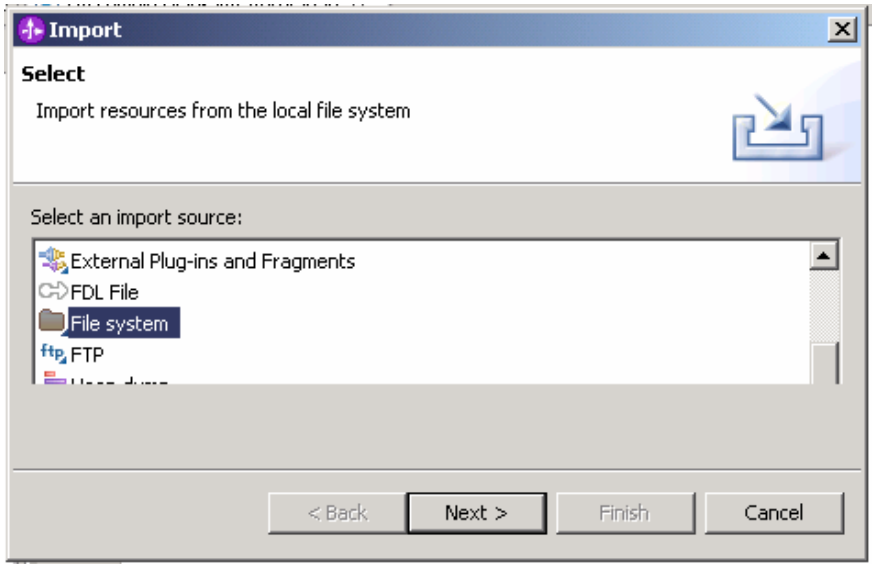
During the first iteration of the cleanse process a simple Java implementation will be created to handle the cleanse process. This means you need a Java implementation to start the process you are going to test. This process involves a Java implementation sending a message through a selector, where the selector chooses to use a POJO 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…**

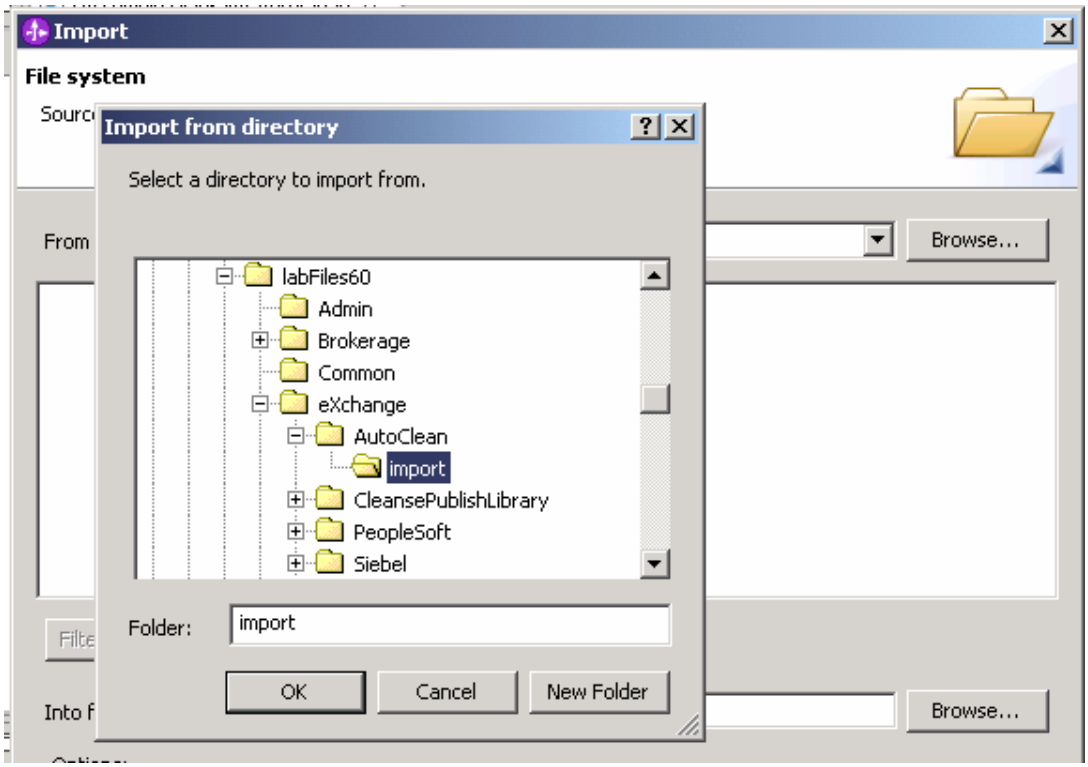


__ b. Select File System, click Next



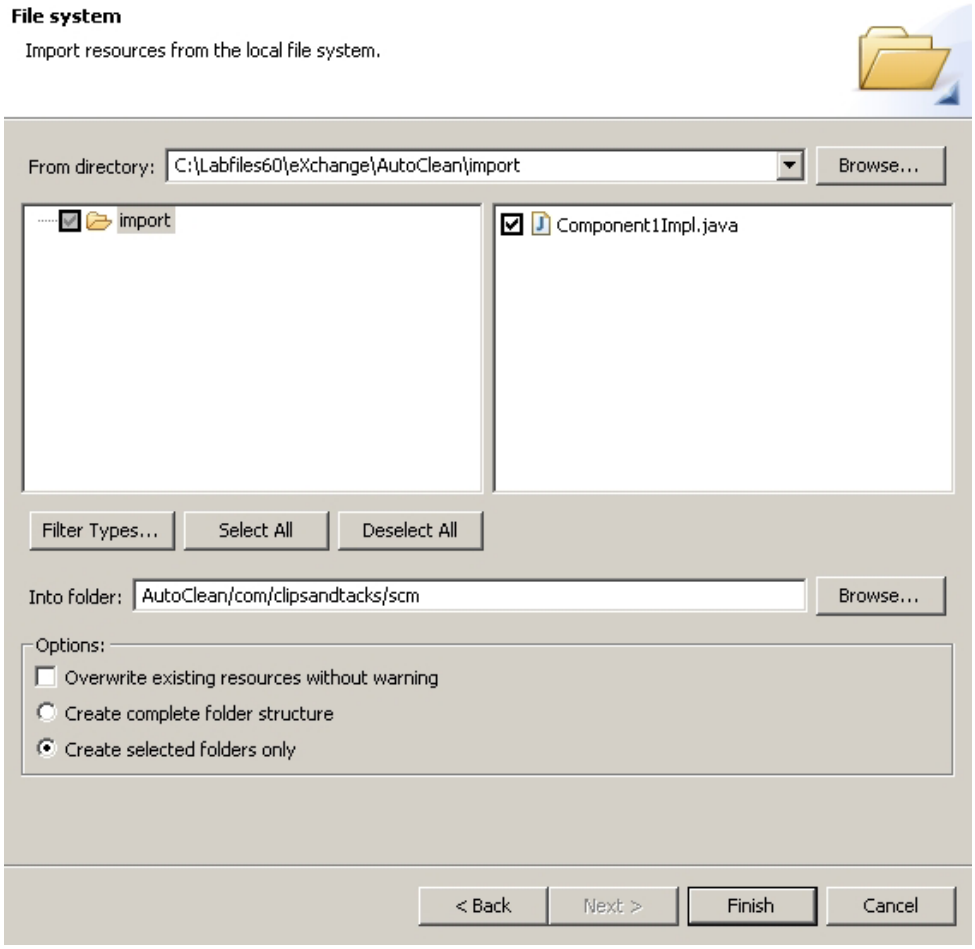
© Copyright IBM Corporation 2005. All rights reserved

__ c. Click Browse, click My Computer, <drive>, labFiles60->eXchange->AutoClean->import



© Copyright IBM Corporation 2005. All rights reserved

__ d. Click the check box for **Component1.java**, enter **AutoClean/com/clipsandtacks/scm** for the Into Folder value, click **Finish**.

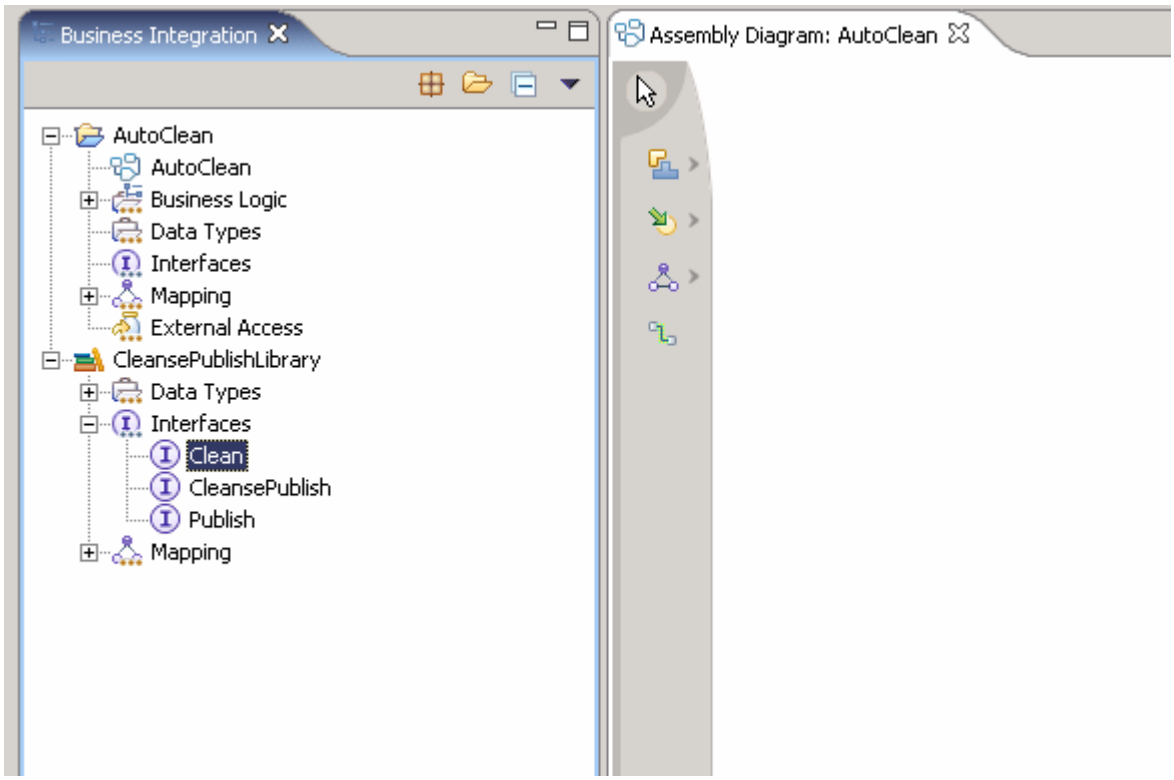


____ 2. Create Java Component

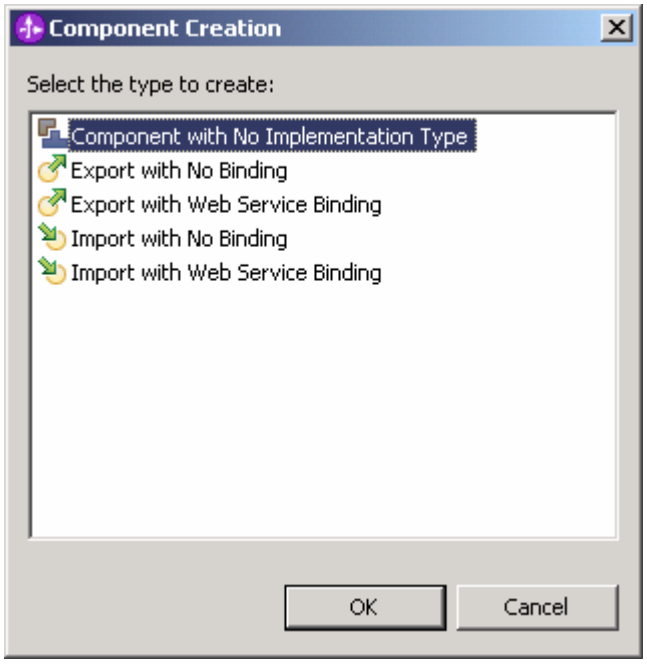
__ a. Double Click AutoClean folder in Business Integration view to start Assembly Diagram: AutoClean

© Copyright IBM Corporation 2005. All rights reserved

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

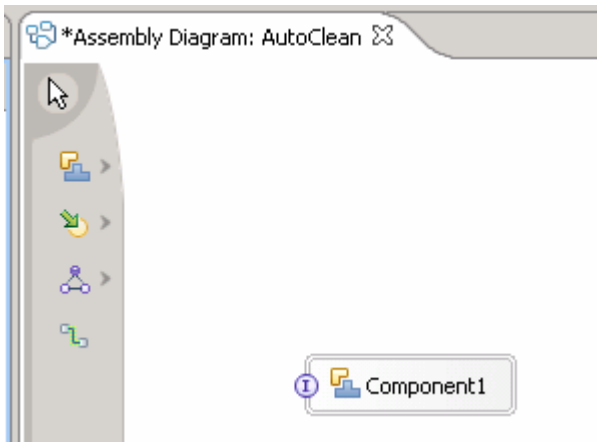


__ c. Accept **Component with No Implementation Type**, click **OK**



© Copyright IBM Corporation 2005. All rights reserved

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

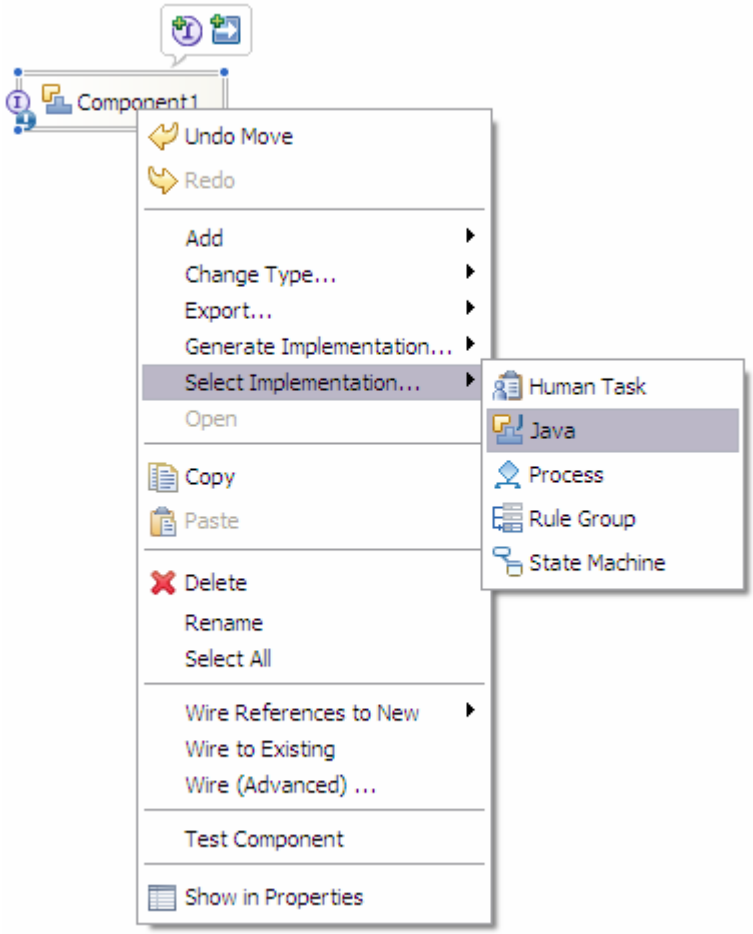


Note: Optionally, the Component1 display name can be changed to a more descriptive name. Right-click, select show in properties, the display name could be edited. For this lab, it is recommended you leave it as Component1

© Copyright IBM Corporation 2005. All rights reserved

____ 3. Pick the Java implementation imported in previous step

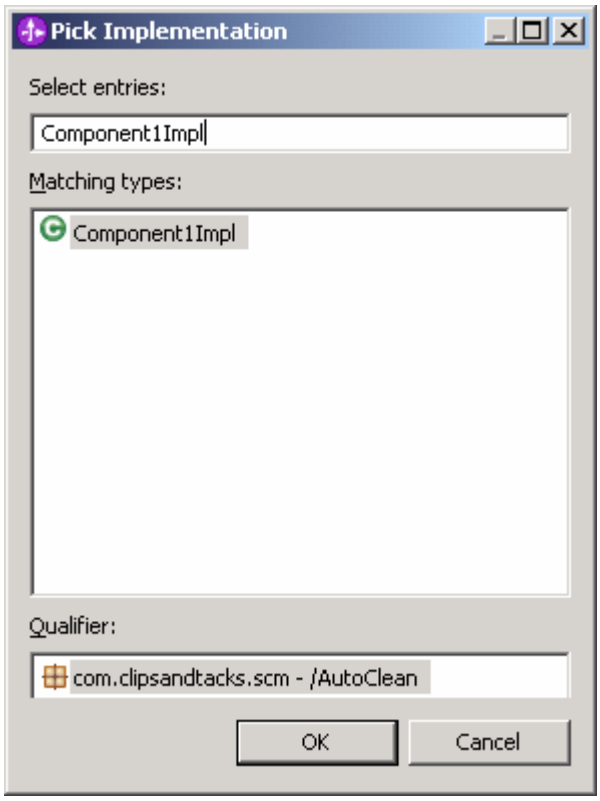
__ a. Right-click on **Component1** in the Assembly Diagram, select **Select Implementation->Java**



Note: Optionally, a new java implementation could be generated using the **Generate Implementation…**

© Copyright IBM Corporation 2005. All rights reserved

__ 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 we have done it for you, and added the appropriate code to use in this lab.

____ 4. Press **<ctrl>-s** to save the workspace. This will also build the workspace.

© Copyright IBM Corporation 2005. All rights reserved

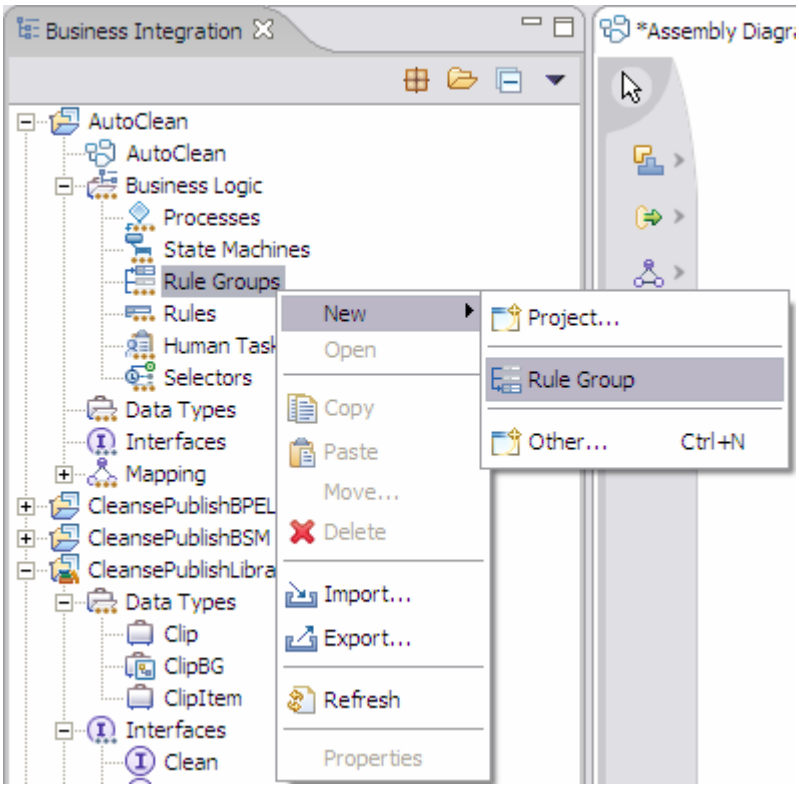
Part 3: Create a Business Rule Group

In this section, you will be creating a Business Rule Group. A Rule Group must be set up before creating a business rule implementation (Ruleset 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.

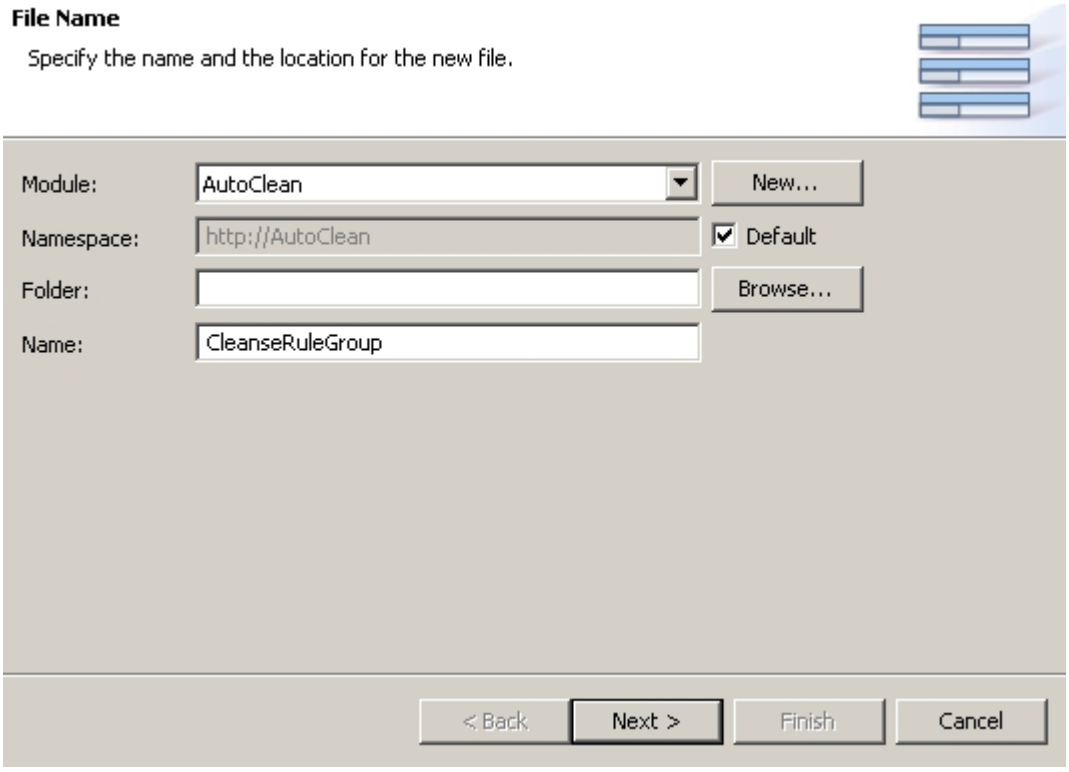
____ 1. Create a Rule Group

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

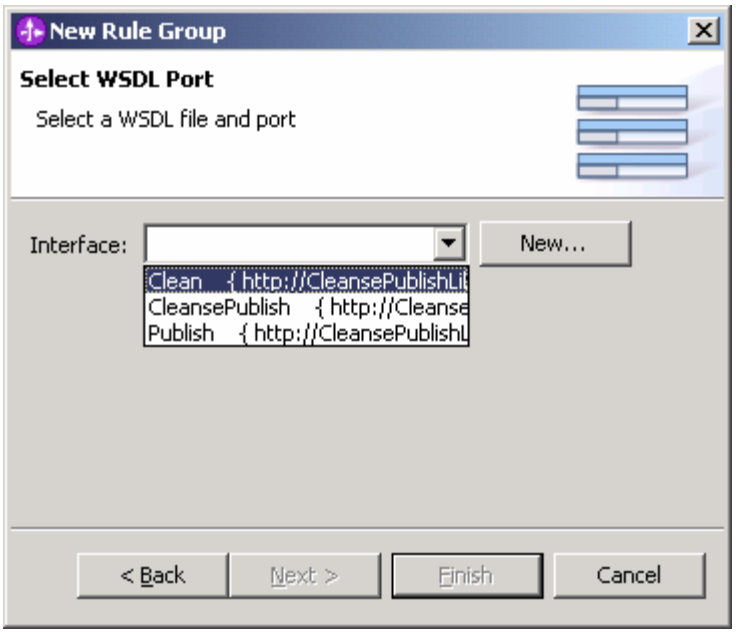


© Copyright IBM Corporation 2005. All rights reserved

__ 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.

© Copyright IBM Corporation 2005. All rights reserved

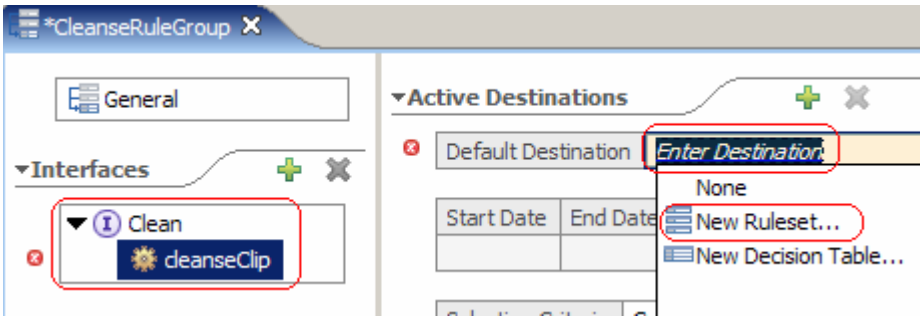
Part 4: Create a Business Rule Set

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.

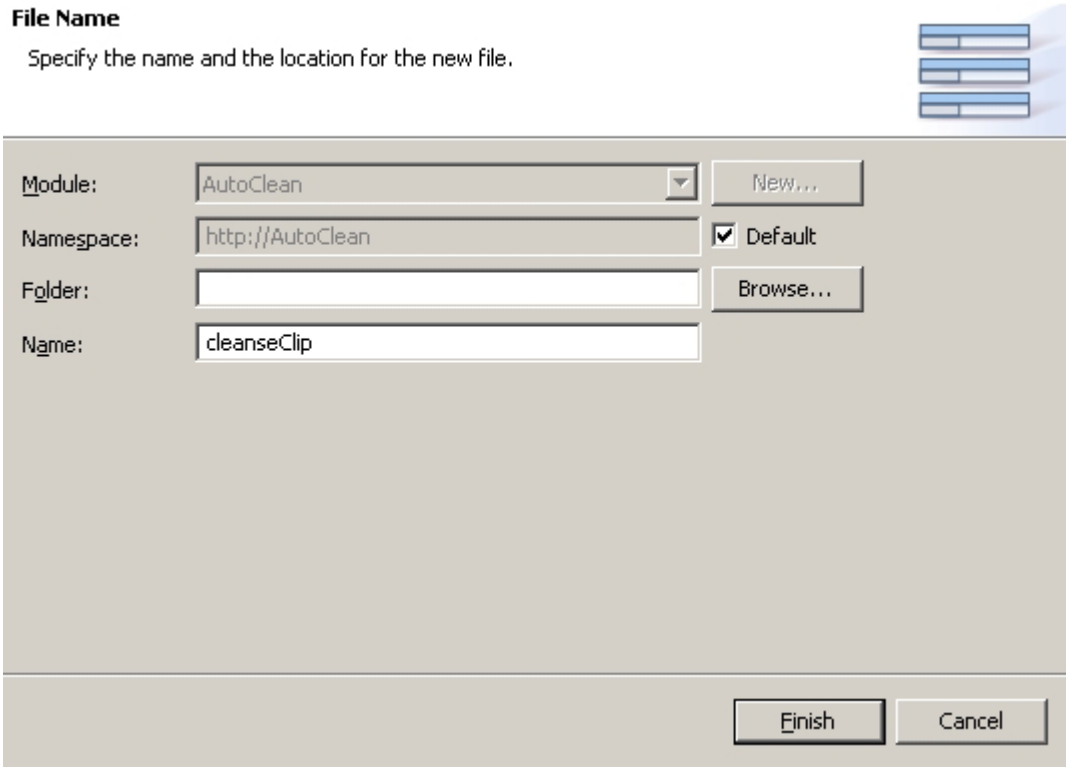
NOTE: You may receive errors/warnings in the Problems view as you create components. These errors/warnings will come and go as you save/build/finish building pieces.

____ 1. Create a Rule Set

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

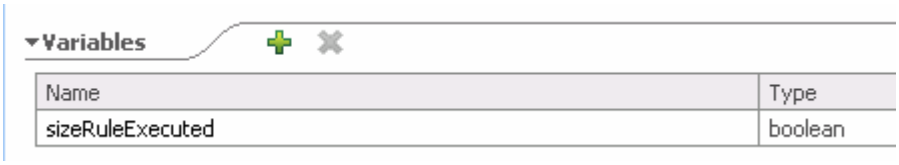


__ b. Accept the default values on the New Ruleset screen, click **Finish.**



© Copyright IBM Corporation 2005. All rights reserved

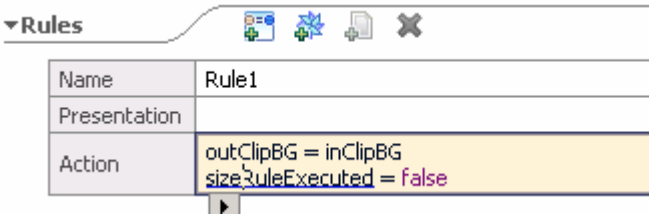
__ c. Click to add a local variable. Enter **sizeRuleExecuted** for the name, **Boolean** for the Type.



__ d. Click 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.

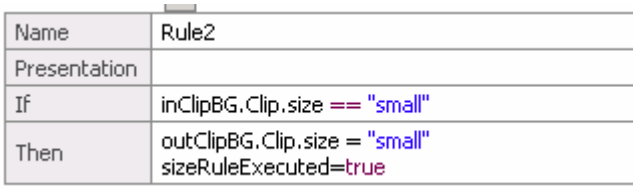
__ e. Click **Enter** at the end of the line to enter the initialization of the sizeRuleExecuted variable. The rule should look as follows. If not, simply delete the text of the Action field and repeat selection.

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



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

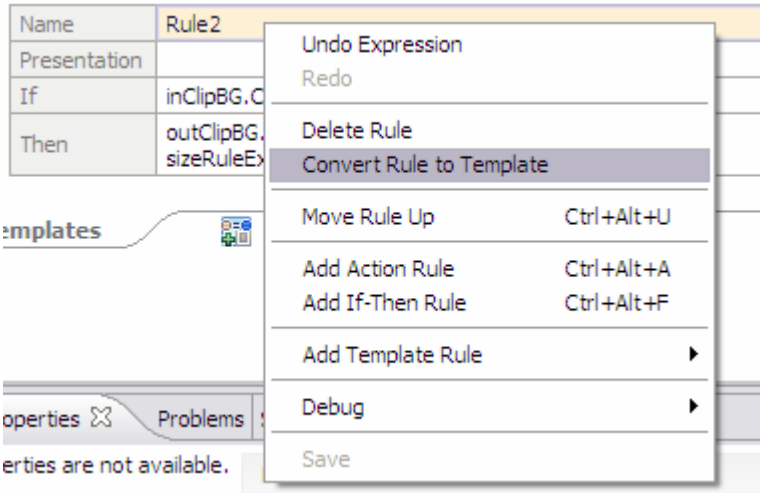
__ f. Click to add an If-Then rule. Click in the If field and select or type **inClipBG.Clip.size**, and **==** from the menu, enter **“small”** for the If field. Click in the Then section and select or type **outClipBG.Clip.size** and **=** from the selection menu, enter **“small”,** hit enter to advance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



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 this Business Rule so later in this lab you can see it from the Business Rule Editor web tool.

© Copyright IBM Corporation 2005. All rights reserved

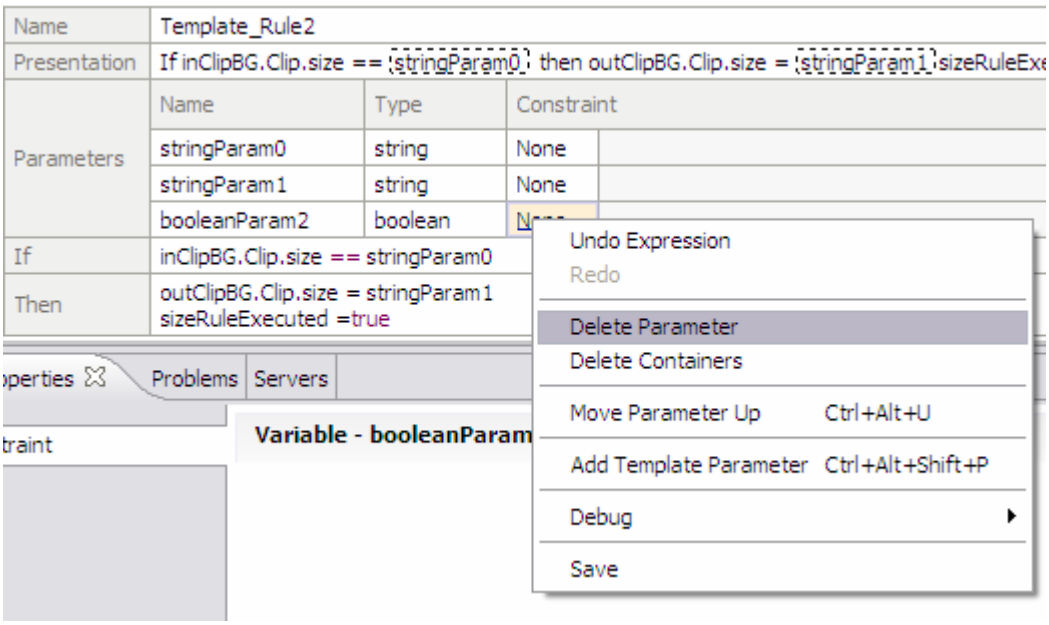
__ g. Right-click on Rule 2, and select **Convert Rule to Template**



__ h. A template will be generated for Rule 2.

__ i. Change the Template_Rule2 “Then” field: sizeRuleExecuted = booleanParam2 to sizeRuleExecuted = true

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



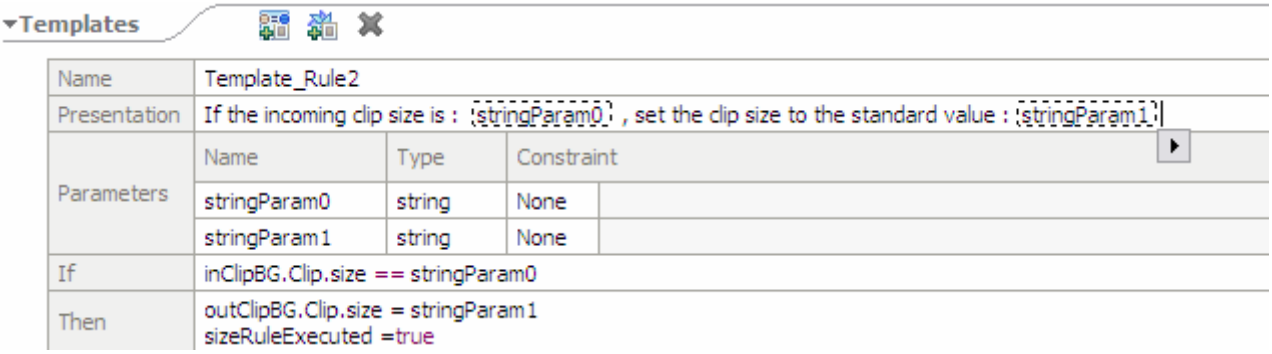
1) The place holder for the Boolean variable {2} should be deleted from the presentation line also (above the Parameters area of template, at end of presentation line). **ALSO** delete the **sizeRuleExecuted=** from the Presentation area.

__ k. 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

© Copyright IBM Corporation 2005. All rights reserved

field in Template_Rule2 to: **If the incoming clip size is : {stringParam0}, set the clip size to the standard value : {stringParam1}.** Delete sizeRuleExecuted= from the Presentation line.

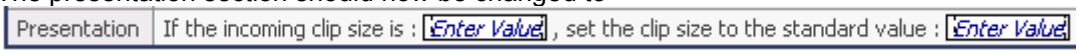
NOTE: Changing the text in the Presentation field is important because this is all the Business Analyst will see when editing the Business Rules in the Business Rule Editor Web tool.



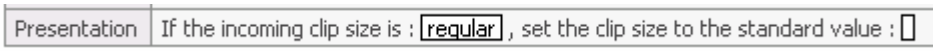
__ l. Creating a few rules from the template

1) Click , and select the only option… Template_Rule2

2) The presentation section should now be changed to



To



3) Repeat the above 2 steps to create **4 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 **medium** and **<a blank>**

c) With values **large** and **large**

d) With values **jumbo** and **large**

__ m. 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**

© Copyright IBM Corporation 2005. All rights reserved

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.

__ n. 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		

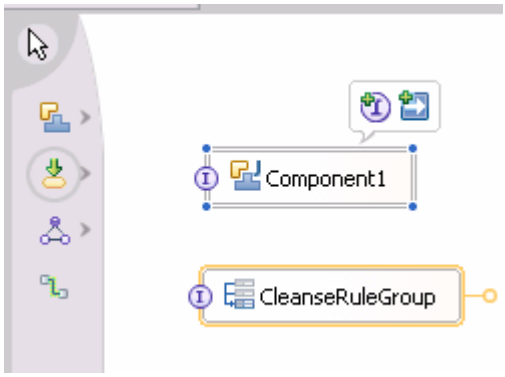
© Copyright IBM Corporation 2005. All rights reserved

__ o. Save the Rule Set by pressing **<ctrl>-s**.

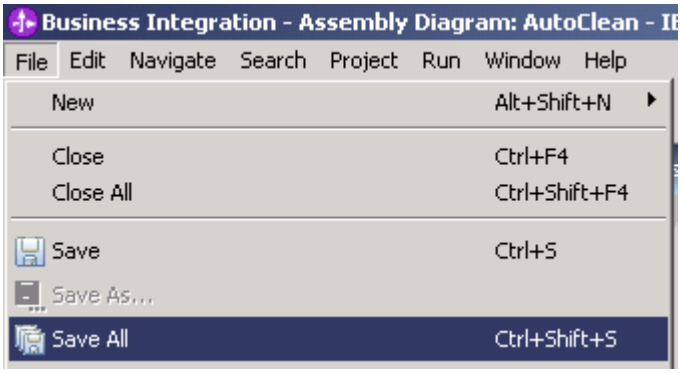
__ p. Add CleanseRuleGroup to the Assembly Editor.

1) Go to the **Assembly Editor** by double clicking the AutoClean folder

2) Drag CleanseRuleGroup onto the assembly editor.



____ 2. Save all, **File->Save All,** this will build the workspace.



© Copyright IBM Corporation 2005. All rights reserved

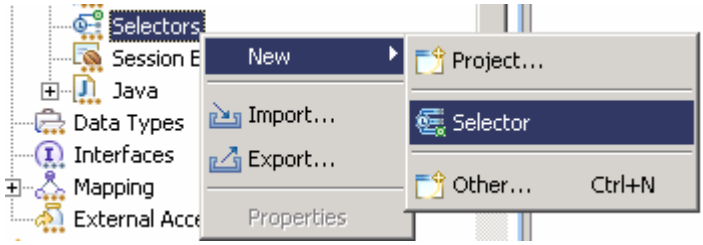
Part 5: Create a Selector

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.

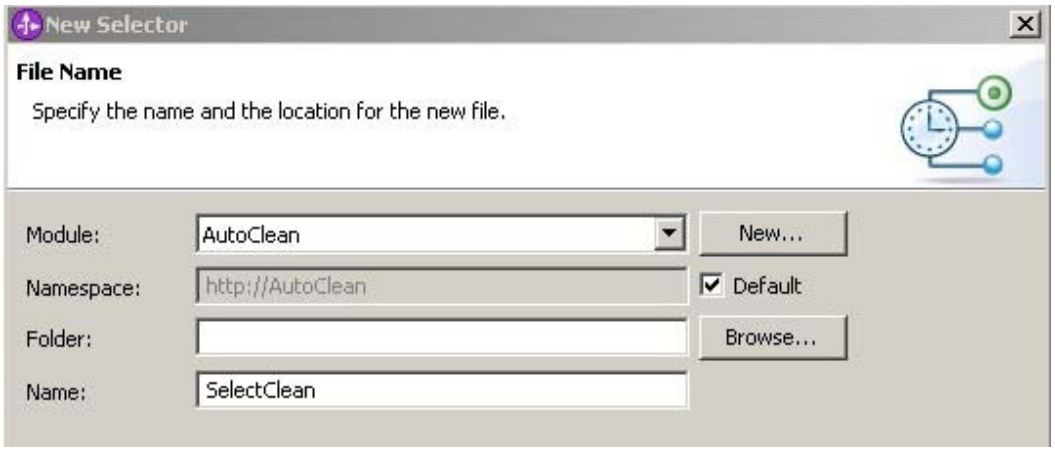
____ 1. Create a new Selector

__ a. Expand the AutoClean Module. Expand Business Logic

__ b. Right click on the **AutoClean -> Business Logic -> Selectors** folder, select **New->Selector**

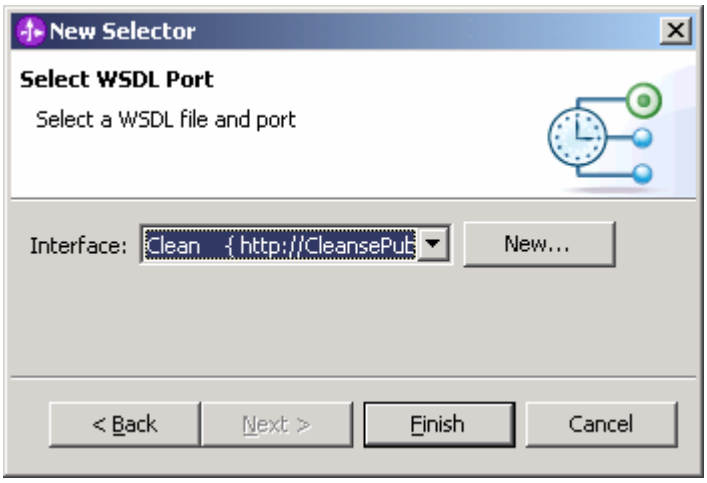


____ 2. Enter **SelectClean** for the Name, click **Next**.



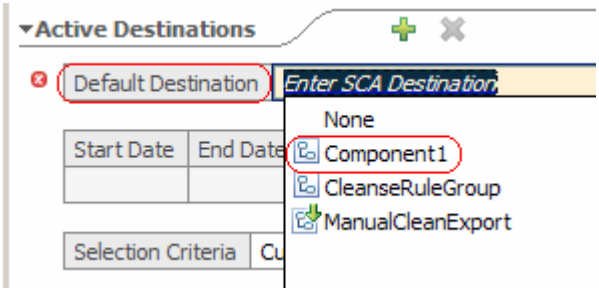
© Copyright IBM Corporation 2005. All rights reserved

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



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

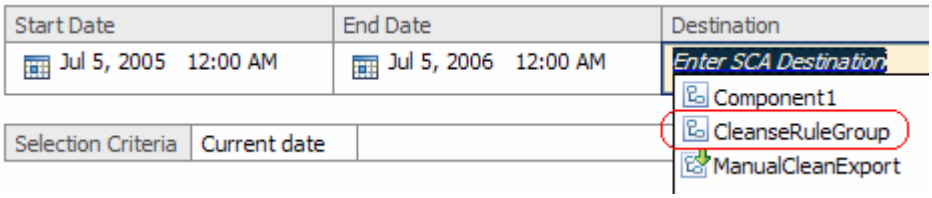
____ 5. Click **Enter SCA Destination** next to Default Destination and select C**omponent1**



____ 6. Click on the **plus icon** to add date selection entry.

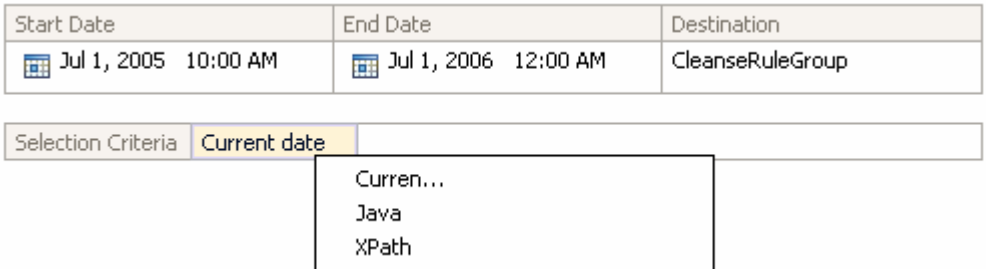
____ 7. Change the time to add **1 hour** to the current time under the Start Date field. You will be adding an hour so that in 1 hour, the selector will switch over from using Component1 (the POJO) to the Business Rule.

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

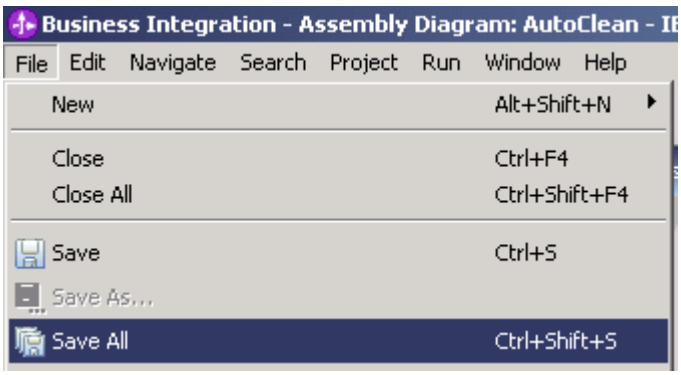


© Copyright IBM Corporation 2005. All rights reserved

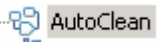
____ 9. Click **Enter Method** for Selection Criteria, select **Current date**

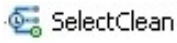


____ 10. Select **File->Save All** to save and build the workspace

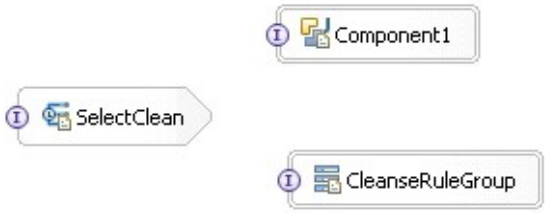


____ 11. Place Selector on Assembly Diagram

__ a. Double Click  folder in Business Integration view to open the **Assembly Diagram** for the AutoClean module.

__ b. Drag  onto the Assembly Editor

__ c. Your Assembly Editor should look like this:



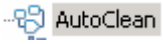
NOTE: You cannot wire the Selector to the destination components. However, the Selector is already wired to the two target components - Component1 (the POJO) and the CleanseRuleGroup (the Business Rule). The selector is using dynamic wires to point at Component1 and the CleanseRuleGroup under the covers. 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.

© Copyright IBM Corporation 2005. All rights reserved

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 

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

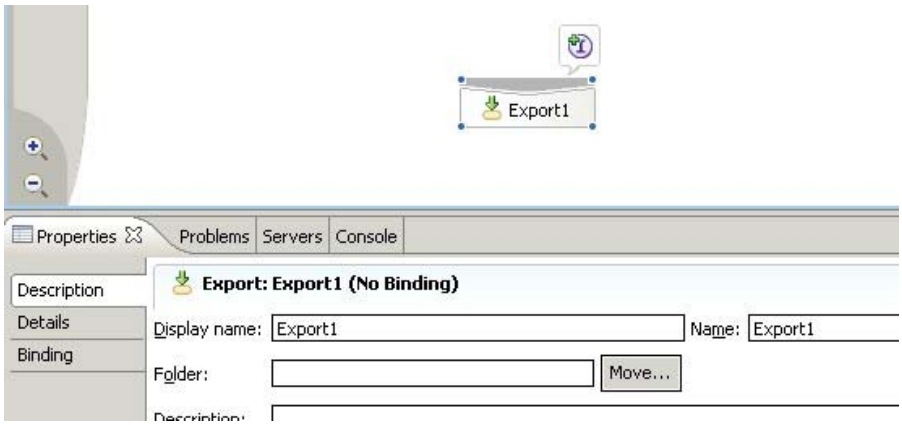
__ a. Open and click the Export button on the Assembly Editor…

__ b. After clicking the button, move the cursor over the Assembly Editor and click inside the Assembly Editor white 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.

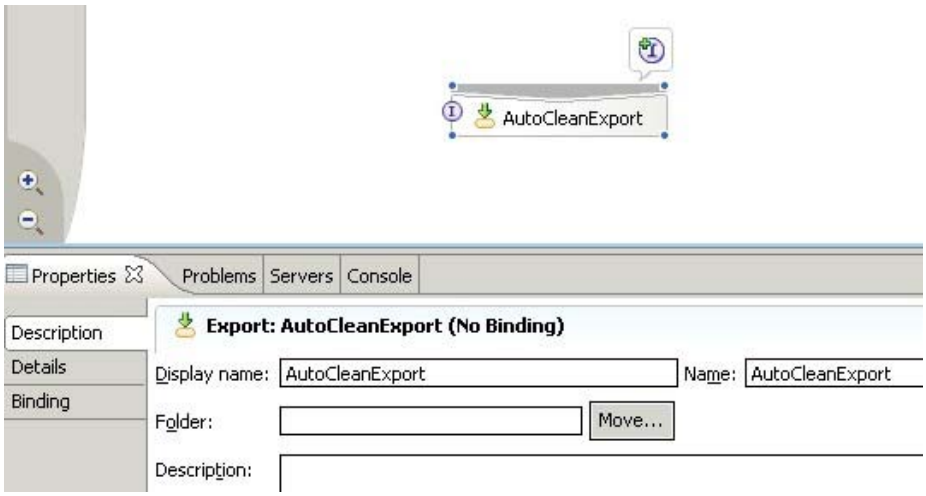
__ a. Click on **Export1** in the Assembly Editor so that it is highlighted with focus. (Hint: click on text in icon for one click selecting)

© Copyright IBM Corporation 2005. All rights reserved



__ b. Open the **Properties View** (found at bottom of Business Integration Perspective along with Servers, Problems, Console)

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



____ 4. Add the Clean interface to the Export. The reason you need to add an interface is to define the inputs and outputs of the module, so requests from clients know what datatypes 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.

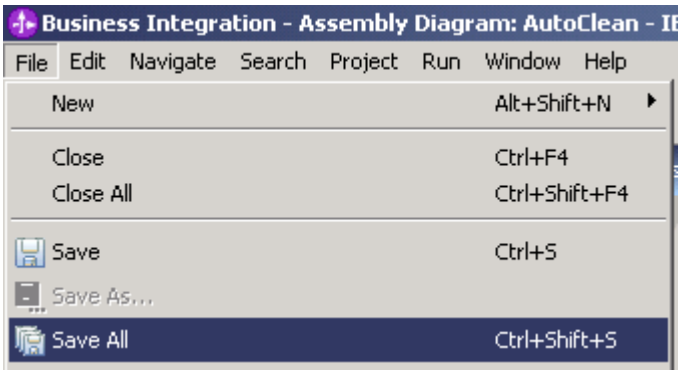
© Copyright IBM Corporation 2005. All rights reserved

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

__ b. Click on AutoCleanExport first, 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.

____ 6. Your Assembly Editor should now look something like the screenshot below.

____ 7. Select **File->Save All** to save and build the workspace.



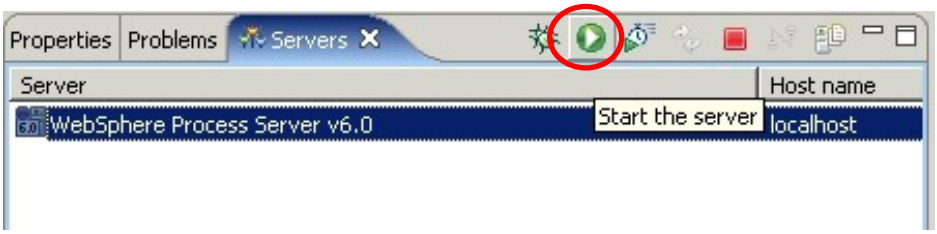
You now have created an export so this Selector/Business Rule module can be accessed and used by the other modules.

© Copyright IBM Corporation 2005. All rights reserved

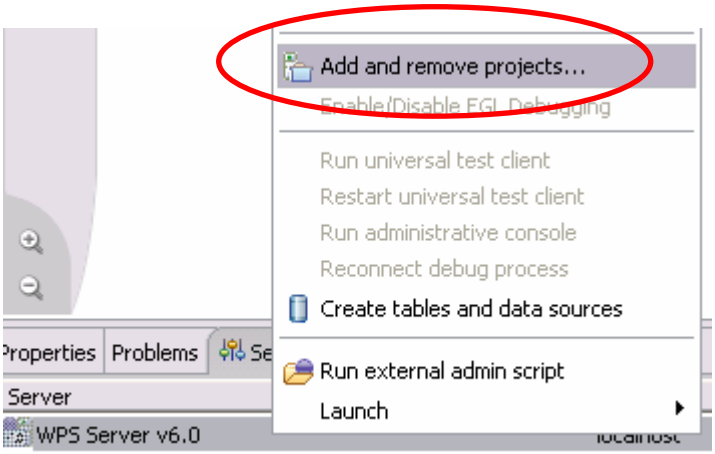
Part 7: Publish the AutoCleanApp

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 tooling there is a full version of the WebSphere Process Server that you can use to auto-publish to. 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 tooling.

____ 1. **Start server** from the Servers View and wait for it to start successfully.

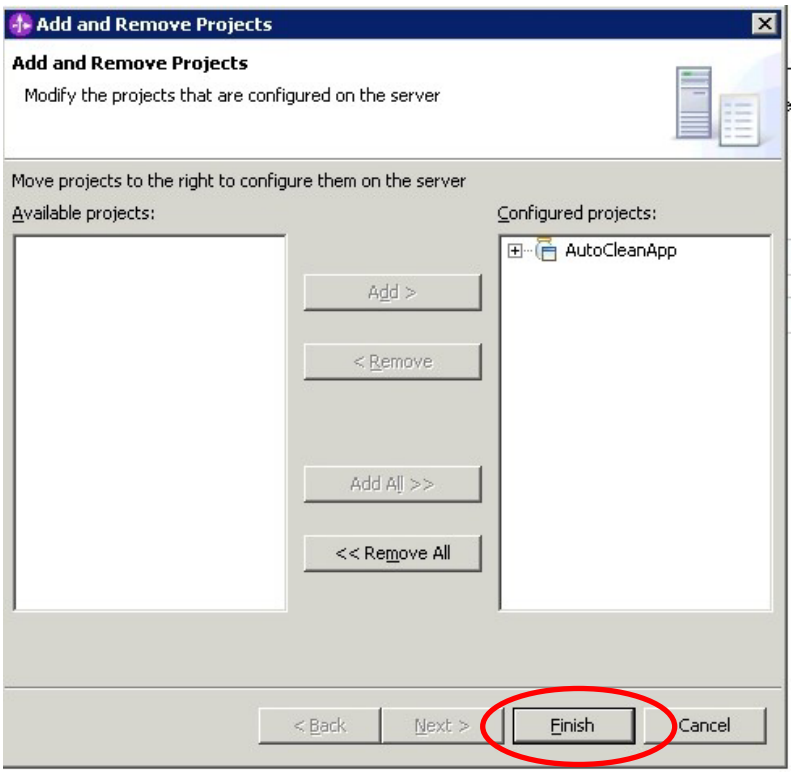


____ 2. Right click on the WPS Server in the Servers View and choose **Add and Remove Projects**



© Copyright IBM Corporation 2005. All rights reserved

____ 3. Click the **Add** button to add the AutoCleanApp to the WPS server. Click **Finish**.



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

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 "templatized" and are running on the WebSphere Process Server or WPS. You can only see Business Rules that have been converted to a template. For instance, Business Rules converted in the tooling or WebSphere Integration Developer (WID) 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 (WPS), and not part of the tooling (WID).

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:

© Copyright IBM Corporation 2005. All rights reserved

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 WAS is set to be disabled
- Code page of command prompt or shell is set to be US environment
- WPS 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

Win32 : 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 app in the Admin Console under Enterprise Applications (application is named BRManager in list of apps).

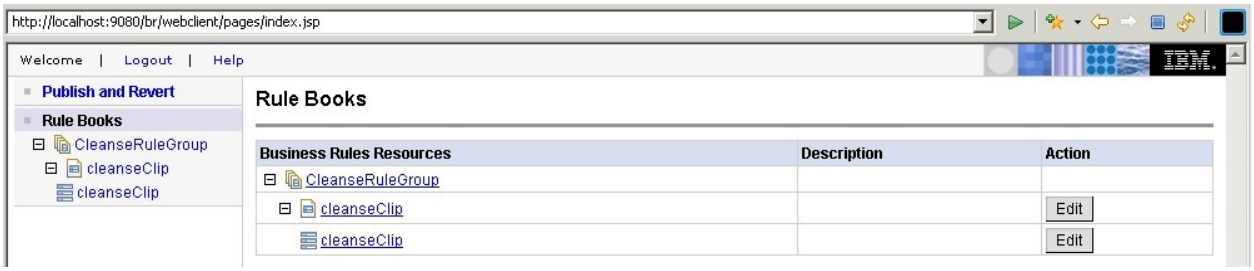
Access the web tool by opening a browser and plugging in this URL
http://localhost:9080/br/webclient/

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

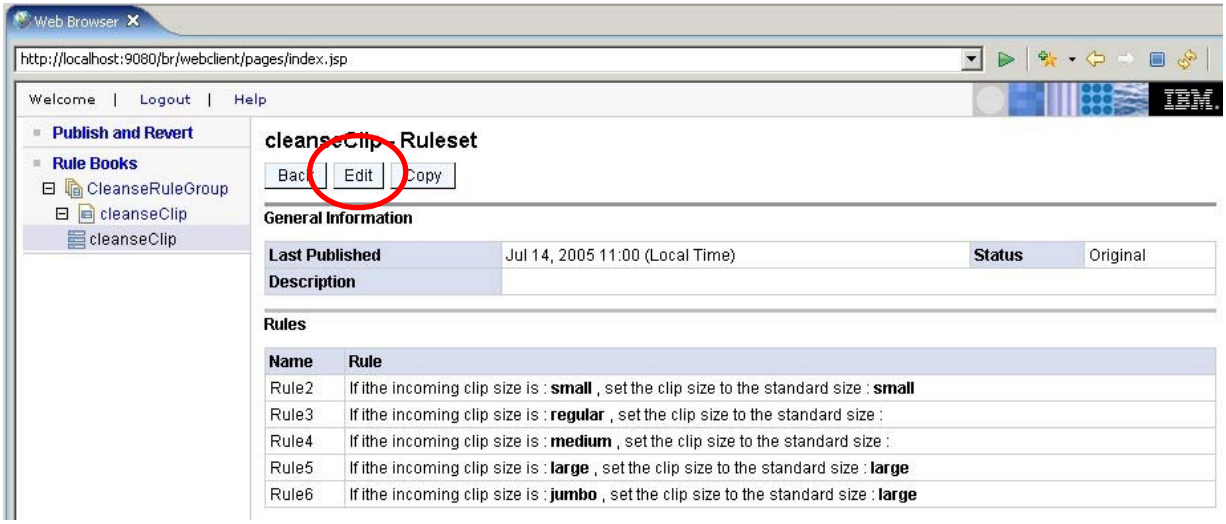
NOTE: If the BRManager does not pick up any business rules at first time, simply restart the server.

____ 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 double clicking the **lower cleanseClip link**.

© Copyright IBM Corporation 2005. All rights reserved

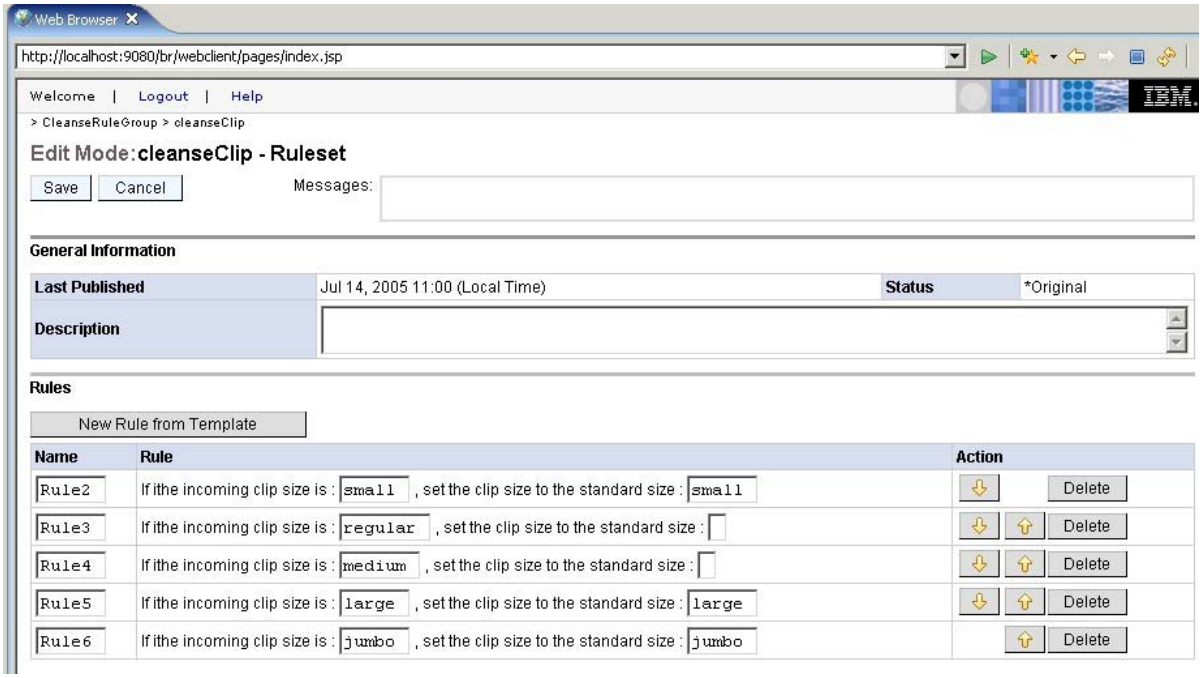


____ 7. These are the rules that were deployed when you published the AutoClean Application. Click the **Edit** button at top to edit the templates you set up

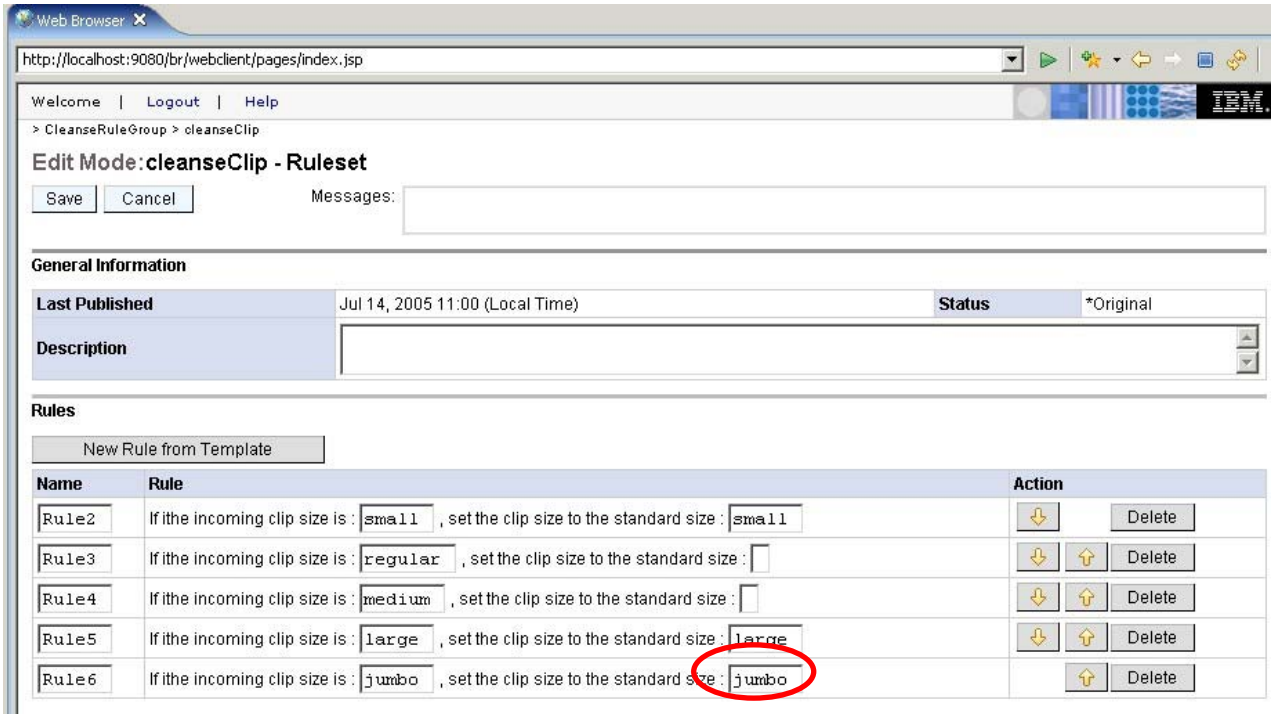


____ 8. With the Web tool, you can change template values, move rules up or down for order of execution (top first, bottom last), and the ability to delete a templatized rule. If the rule is not templatized, you or a business analyst will not be able to see those rules in the Web tool.

© Copyright IBM Corporation 2005. All rights reserved

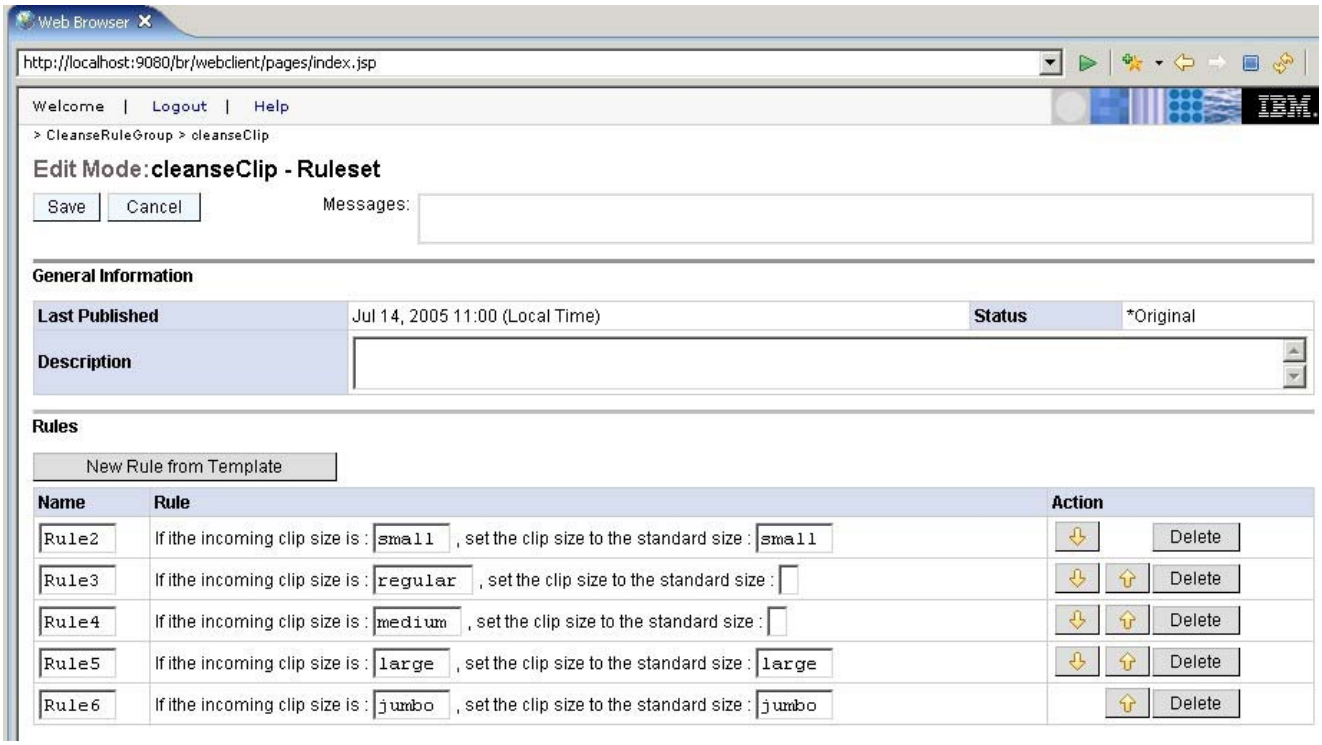


____ 9. Go to the rule with Jumbo and change the result string from **large** to **jumbo**

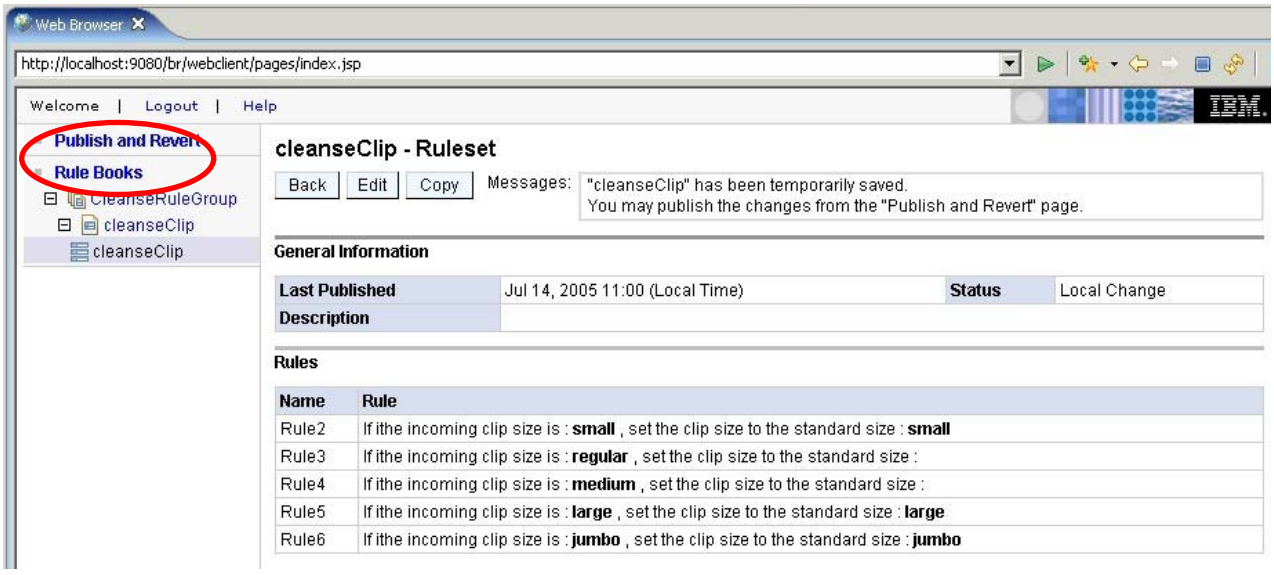


____ 10. Click the **Save button**

© Copyright IBM Corporation 2005. All rights reserved

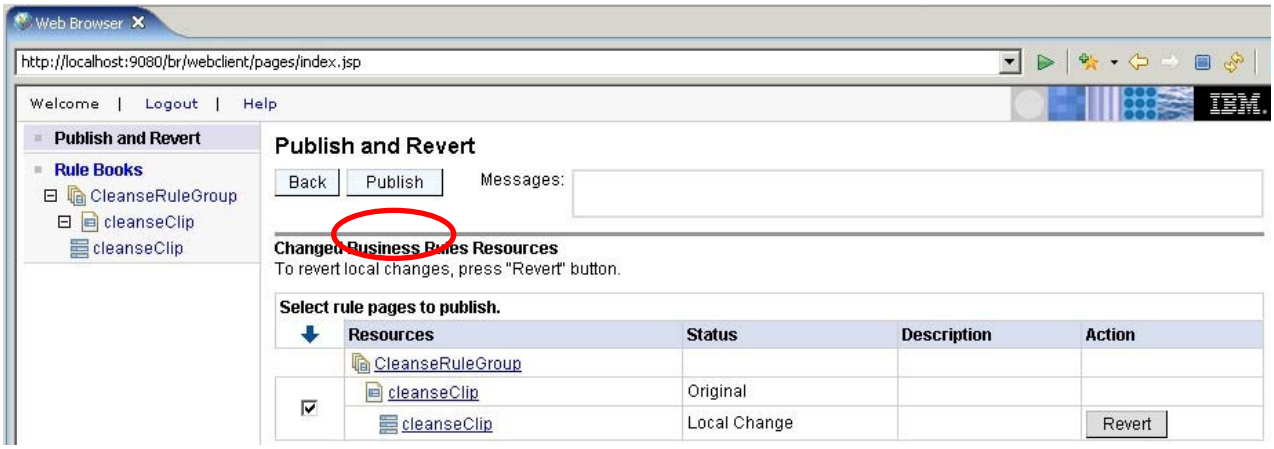


____ 11. Click the **Publish and Revert** link to publish the changes to the application server



© Copyright IBM Corporation 2005. All rights reserved

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



Now, if we were running this application on a production server the input “jumbo” would produce “jumbo”, instead of “large” like we 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.

© Copyright IBM Corporation 2005. All rights reserved

Part 8: Test the Business Rule using Component Test

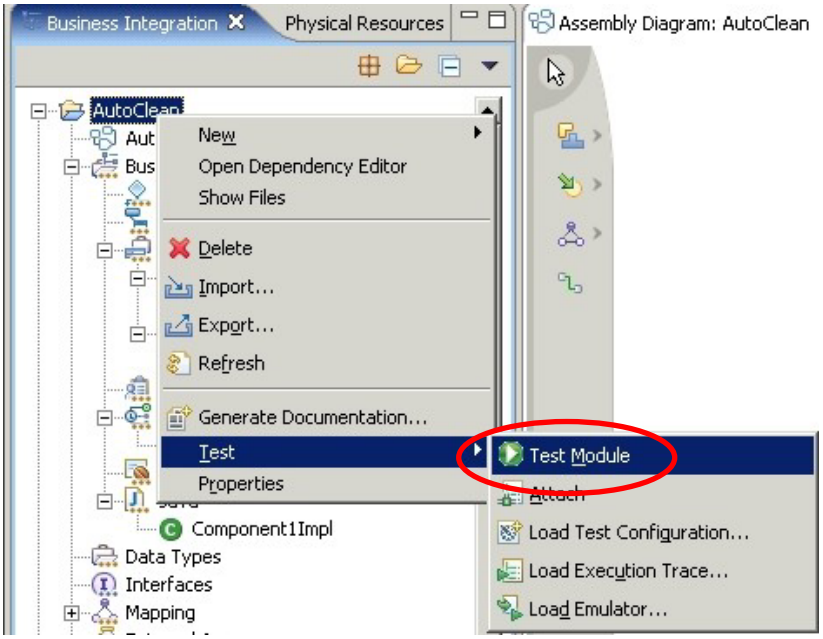
After wiring the components together, its time to test and see if the business rule works. In this section, you will learn to test a business integration module using the component test feature of WebSphere Integration Developer v6.0.

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, top most folder) and chose 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.

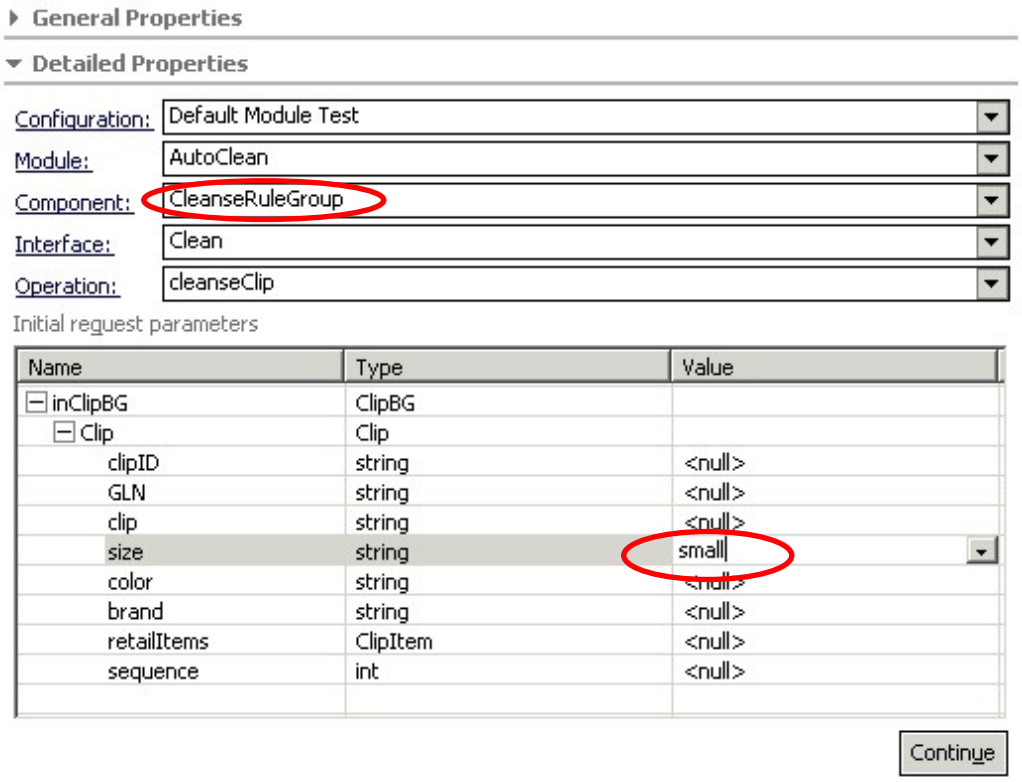
__ b. Look under detailed properties section of the Component Test

Under configuration, you should have **Default Module Test**.
For Module, you should have **AutoClean**
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.
The interface you will be using is the **Clean** interface, and the operation you are testing is **cleanseClip**.

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

© Copyright IBM Corporation 2005. All rights reserved

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



__ c. 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.

__ d. 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 tooling)?

____ 3. Clean up test environment.

__ a. In the Servers view, right-click on WPS Server v6.0 and select **Add and remove projects…**.

__ b. Select AutoCleanApp and click **< Remove**. Click **Finish**.

__ c. Right-click on **WPS Server v6.0** and select **Stop**

© Copyright IBM Corporation 2005. All rights reserved

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.

© Copyright IBM Corporation 2005. All rights reserved

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_PI.zip

<MODULE>

n/a

<DEPENDENT_LIBRARIES>

n/a

____ 2. Continue with **Part 7: Publish the AutoCleanApp**