Maintaining Relationships Between ASBOs

What this exercise is about

This exercise shows you how to develop and test relationships. These are used to cross-reference the PeopleSoft ASBOs with the GBOs and from the GBOs to the Siebel ASBOs.

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®)
- IBM WebSphere Business Integration Toolset installed

What you should be able to do

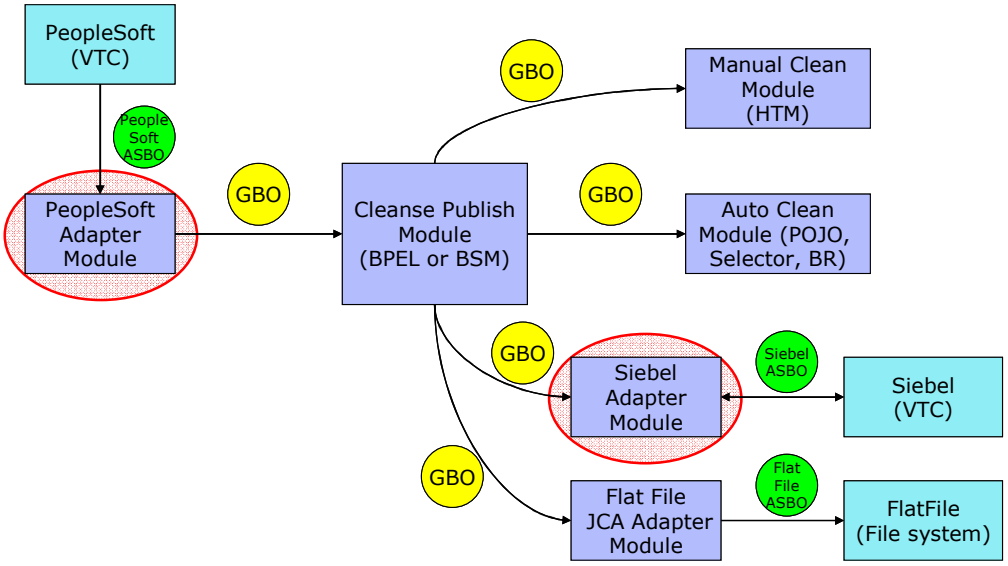
At the end of this lab you should be able to

- Design, test and utilize relationships
    - Create a relationship definition and relationship roles

- Test the relationships using Test Component

Introduction

The following diagram highlights the parts of the overall scenario that will be addressed in this lab. You will add mapping of business objects to the PeopleSoft and Siebel modules. This will build upon what you have already developed for these modules in the **WBI Adapters – Mapping** lab.



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\WBIAdapters\workspaceRel	/tmp/Labfiles60/eXchange//WBIAdapters\workspaceRel
<TEMP>	C:\temp	/tmp

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 WebSphere Integration Developer. 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

This lab exercise is dependent upon the completion of the **WBIAdatpers – Mapping Between ASBOs and GBPS** lab exercise. There are two approaches you can take to performing this lab.

1. **Import pre-built projects** – Using this approach, you will import a project which will establish your workspace with the required constructs from the previous lab exercise.

2. **Use your existing workspace** – Using this approach, you will continue to use the workspace you were using in the previous lab exercise.

It is recommended that you use approach 1 where you import pre-built projects. This is because it prevents possible problems that might have been introduced when you completed the previous lab. In addition, the testing of the previous lab leaves some testing specific constructs that are not needed in this lab.

Perform **only one** of the following steps depending upon your choice of workspace.

If you want to **import pre-built projects**, follow the directions below to initialize the Workspace using the following values:

<WORKSPACE>

C:\Labfiles60\eXchange\WBIAdapters\workspaceRels

<PROJECT_INTERCHANGE>

C:\Labfiles60\eXchange\WBIAdapters\Solution\Mapping_PI.zip

<MODULE>

n/a

<DEPENDENT_LIBRARIES>

n/a

If you want to **use your existing workspace**, follow the directions below to initialize the Workspace using the following values:

<WORKSPACE>

C:\Labfiles60\eXchange\WBIAdapters\workspaceMaps

<PROJECT_INTERCHANGE>

n/a

<MODULE>

n/a

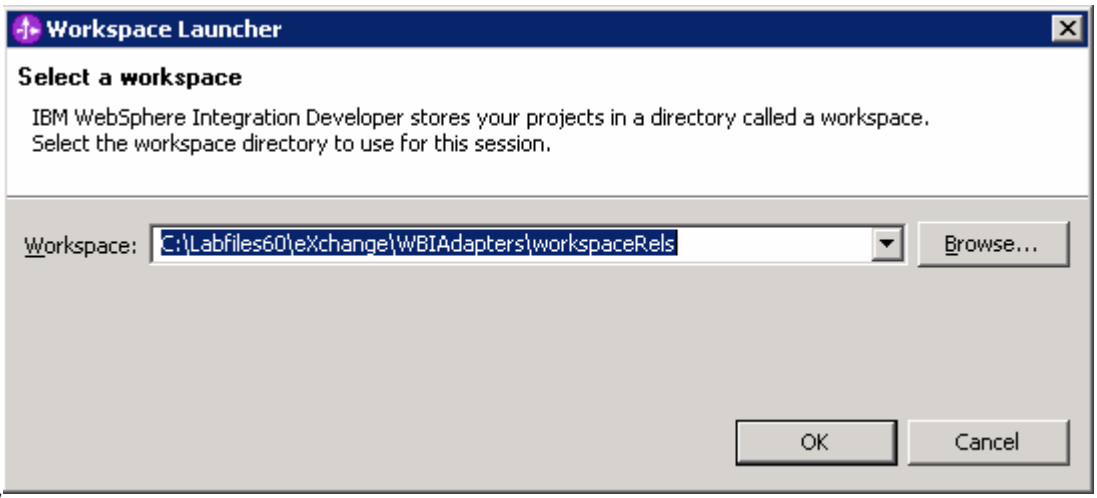
<DEPENDENT_LIBRARIES>

n/a

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.**



…¿

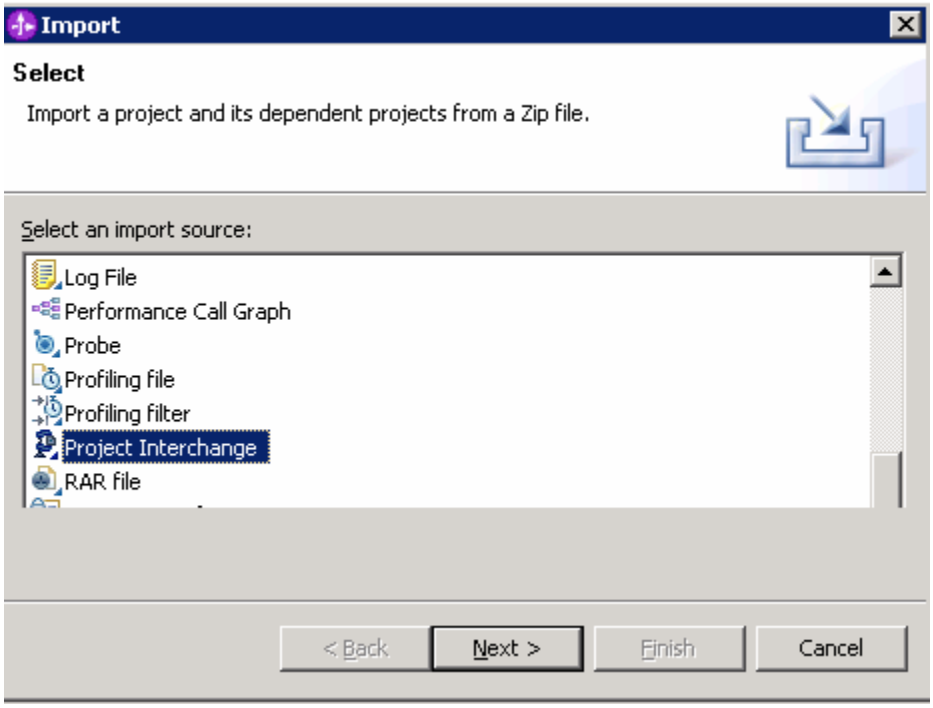
__ c. When WebSphere Integration Developer V6.0 opens, close the **Welcome page** by clicking on the Go to the workbench icon (bent over arrow at top-right).

__ d. Ensure you are in the **Business Integration** perspective.

__ e. If this lab requires you to import a project interchange file, set up the required libraries and modules for this lab by importing the project interchange file **<PROJECT_INTERCHANGE>.**

__ f. Select **File -> Import…** from the menu bar.

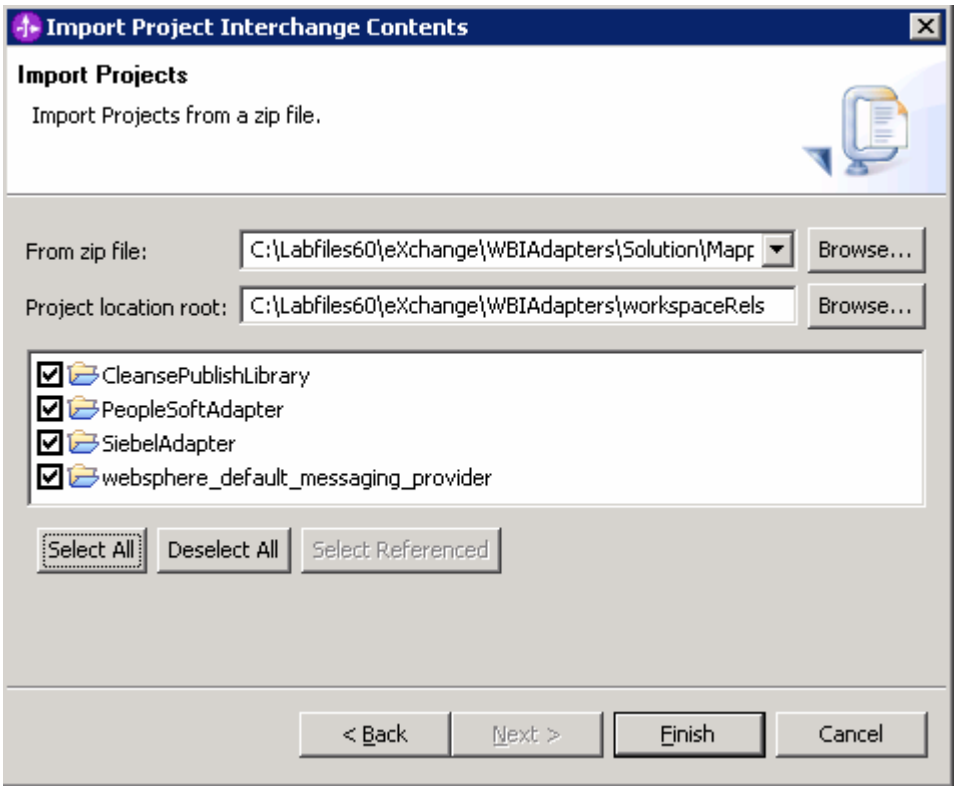
__ g. Scroll down and select **Project Interchange** in the Import dialog.



__ h. Click **Next.**

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

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

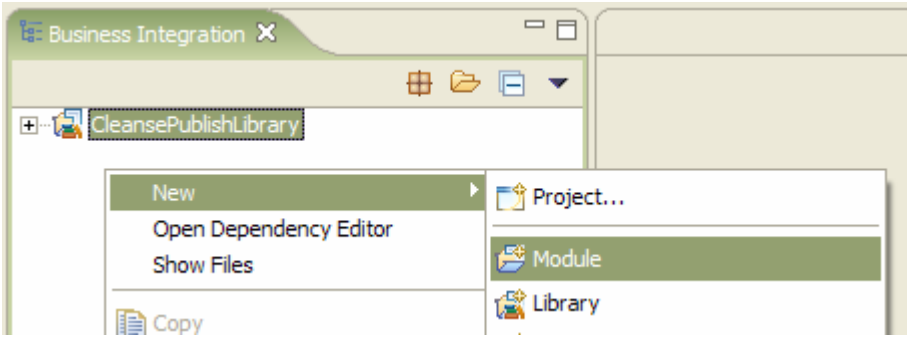


__ k. Click **Finish.**

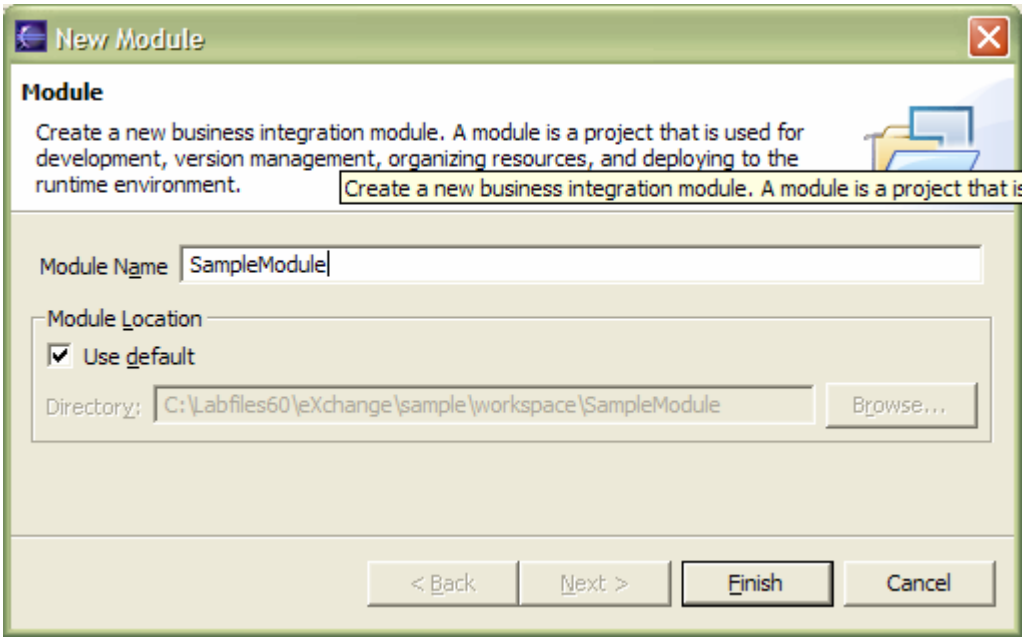
If this lab requires that you create a Business Integration module called **<MODULE>**, complete these steps:

__ l. Right click on the background of the Business Integration view to access the pop-up menu.

__ m. Select **New > Module.**



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

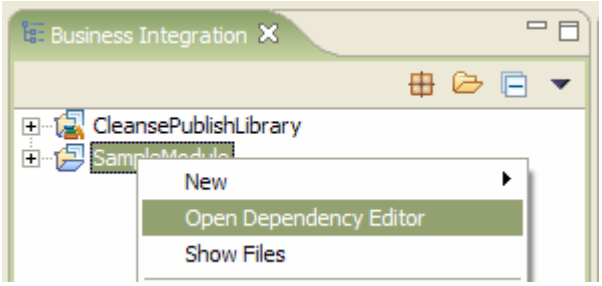


__ o. Click **Finish.**

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

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

__ q. Select **Open Dependency Editor.**



__ r. In the Dependency Editor, click the **Add…** button.

__ s. Select from the **<DEPENDENT_LIBRARIES>** list in the Library Selection dialog.

__ t. Click **OK.**

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

Part 2: Authoring Relationships

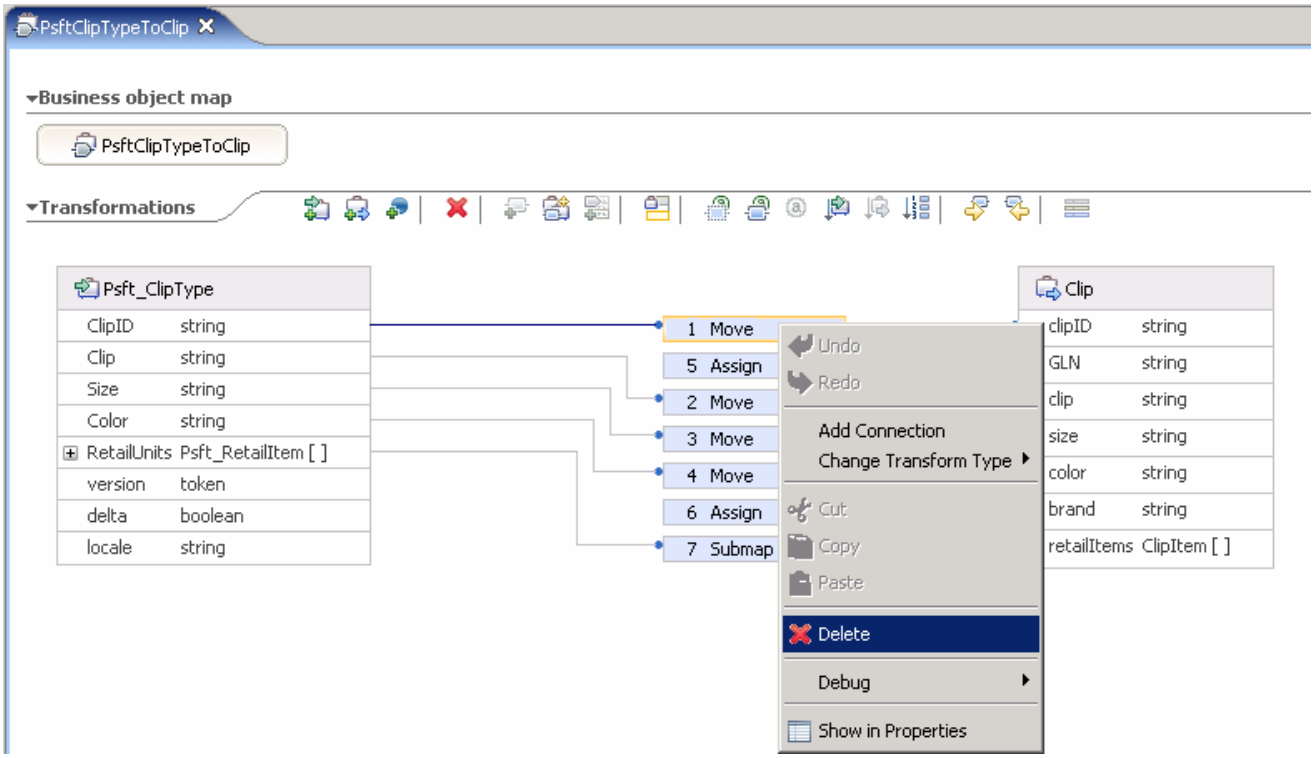
In this section you will modify the maps created in the previous lab to contain an identity relationship that will associate a clip type from PeopleSoft with a product type in Siebel. Logically, the relationship will be between the ClipID field in Psft_ClipType and the Key field in Siebel_ProductType. To do this, the relationships are actually tied together using the clipID from the Clip generic business object and defining it as managed. This then becomes the link through which the Relationship Service ties together ClipID in Psft_ClipType and Key in Siebel_ProductType.

____ 1. Create a relationship between ClipID in Psft_ClipType and clipID in Clip. To do this, you must change the already existing Move transformation for these fields to a Relationship transformation.

__ a. In the Business Integration view, expand **PeopleSoftAdapter > Mapping > Data Maps.**

__ b. Double-click on **PsftClipTypeToClip** to open it in the Mapping Editor.

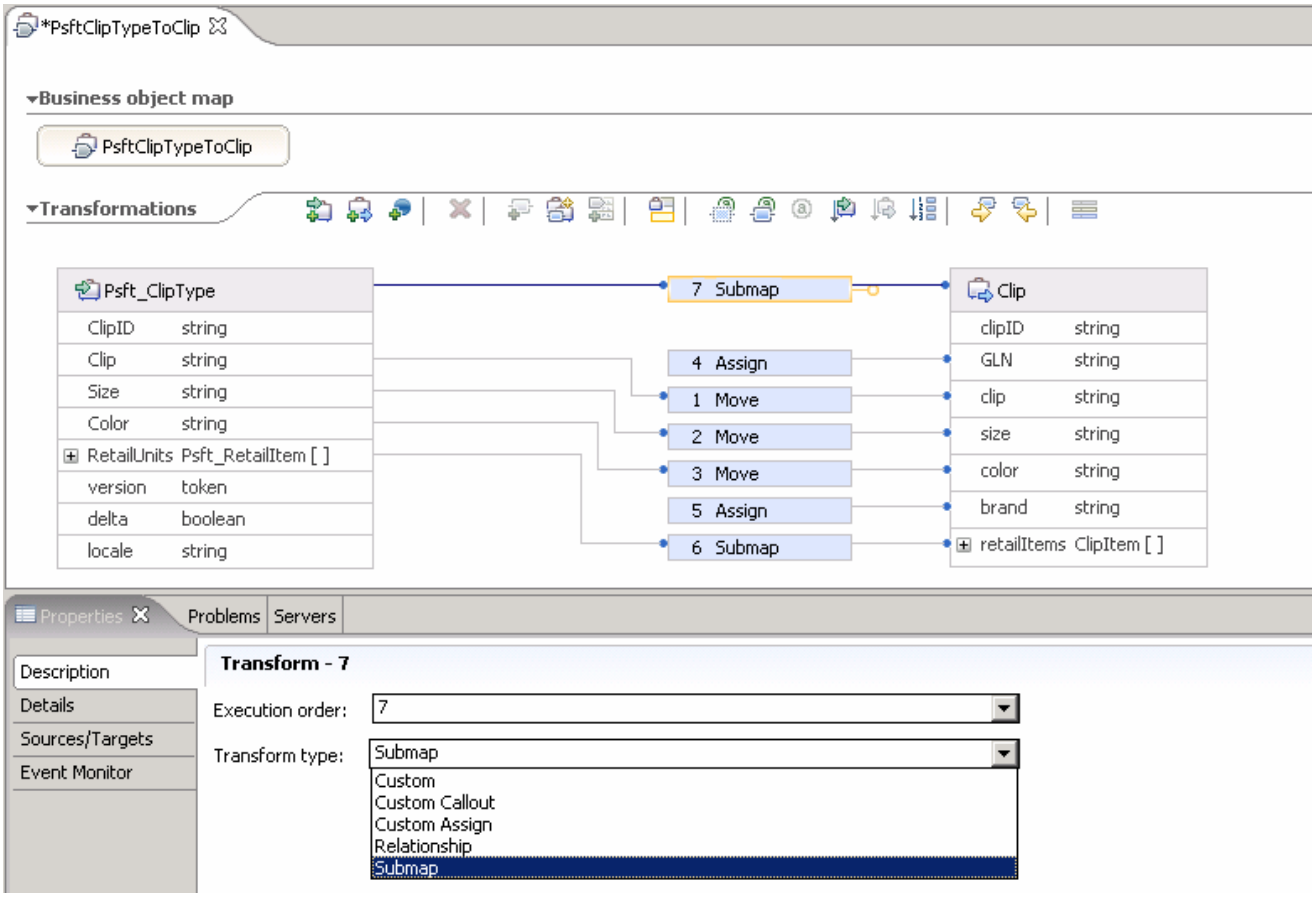
__ c. Right click on the Move between ClipID and clipID and select **Delete.**



__ d. Draw a line from **Psft_ClipType** to **Clip.**

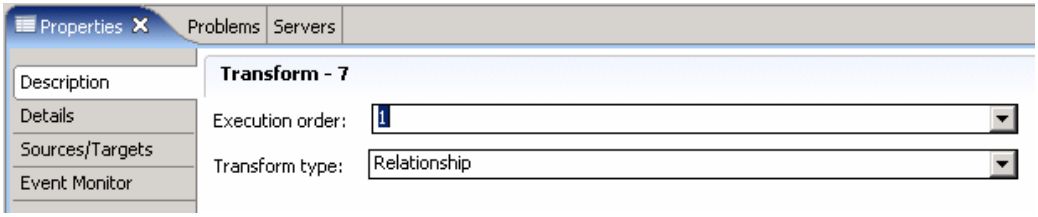
__ e. Click on the left side of the **Submap** transformation rule.

__ f. Click on the **Properties** tab in the lower panel and select the **Description** tab.



__ g. Using the **Transform type** dropdown, change the transformation rule setting from **Submap** to **Relationship.**

__ h. Expand the **Execution order** dropdown and change its setting to **1** for the Relationship rule.



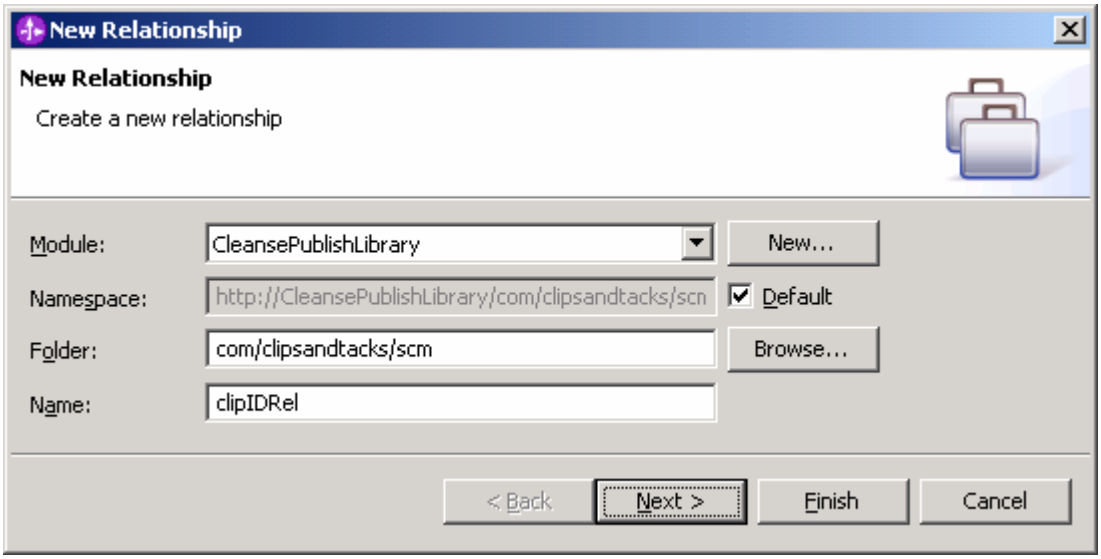
____ 2. Click on the **Details** tab.

__ a. Click the **New…** button by the Relationship Definition field.

__ b. Select the **CleansePublishLibrary** from the **Module** dropdown in the New Relationship prompt.

__ c. Type in a **Name** for the new relationship definition: **clipIDRel.**

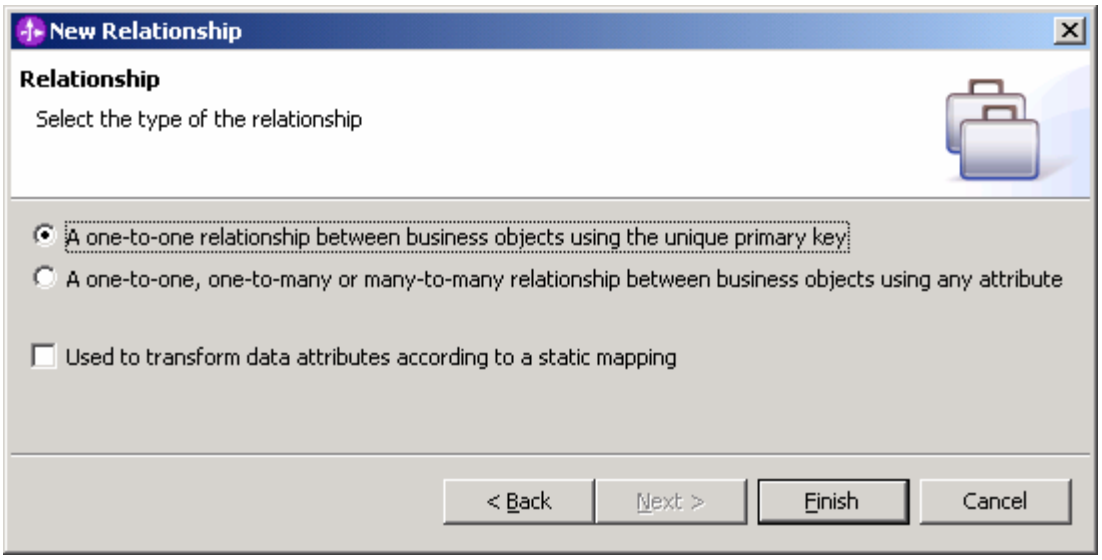
__ d. Ensure that all remaining fields are filled in as shown in the following screen capture.



__ e. Click **Next.**

__ f. Ensure that the **one-to-one relationship** radio button is selected on the next screen.

__ g. Click **Finish.**



__ h. Press **Ctrl+S** to save the new relationship definition, but do not close the relationship definition yet.

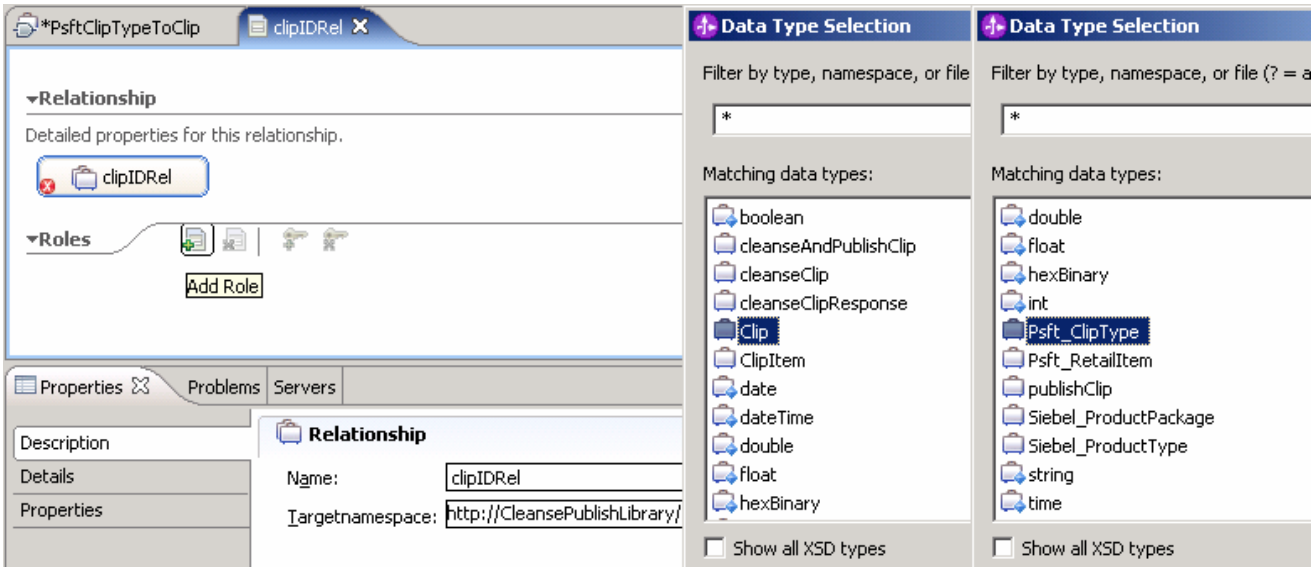
__ i. You might notice that the relationship is now flagged with an error. This is because no Relationship Roles have been defined yet. This error will be resolved shortly.

____ 3. Define the roles in the relationship.

__ a. If you closed the **clipIDRel** component, navigate to **CleansePublishLibrary > Mapping > Relationships** in the Business Integration view and double click on **clipIDRel** to open it.

__ b. Add a new relationship role by clicking on the **Add Role** icon.

__ c. In the Data Type Selection dialog, select **Clip** and click **OK.**



____ 4. Click on the **Add Role** icon again.

__ a. Select **Psft_ClipType** in the Data Type Selection dialog.

__ b. Click **OK.**

__ c. Press **Ctrl+S** to save the relationship definition.

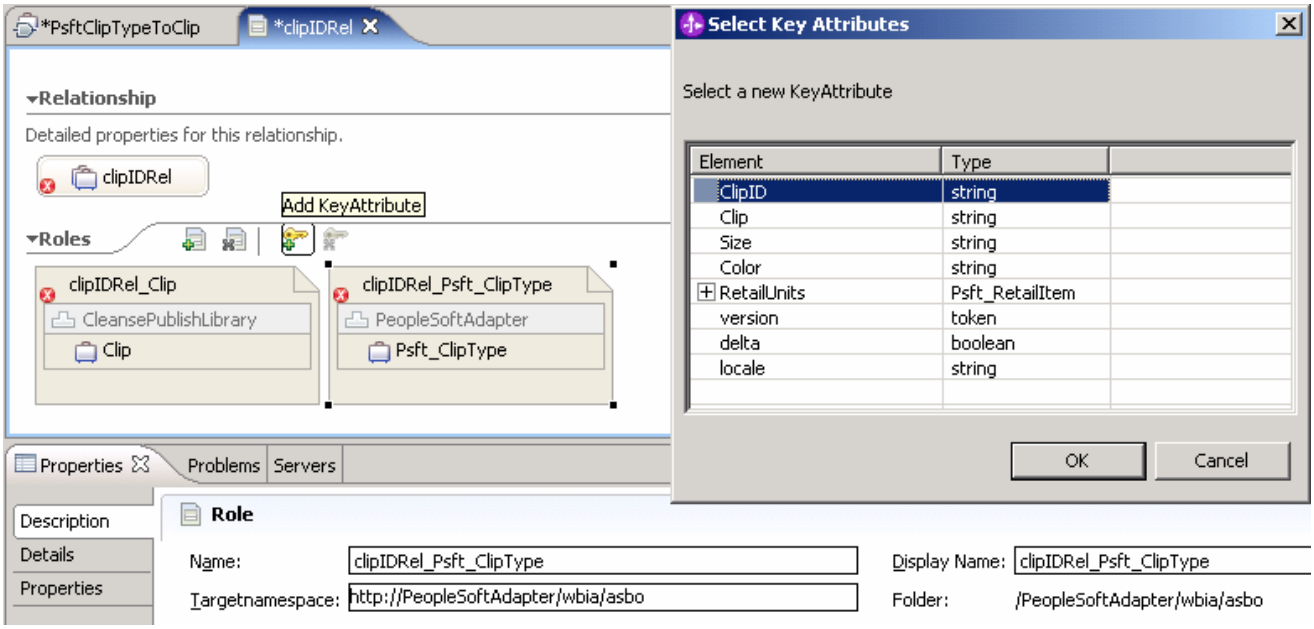
__ d. Click on the **clipIDRel_Psft_ClipType** box to select it.

__ e. Notice the 4 dots at the corners, indicating it is selected.

__ f. Click on the **Add Key Attribute** icon.

__ g. Select **ClipID** in the Select Key Attributes dialog.

__ h. Click **OK.**



__ i. Click on the **clipIDRel_Clip** box to select it.

__ j. Click on the **Add Key Attribute** icon.

__ k. Select **clipID** in the Select Key Attributes dialog.

__ l. Click **OK.**

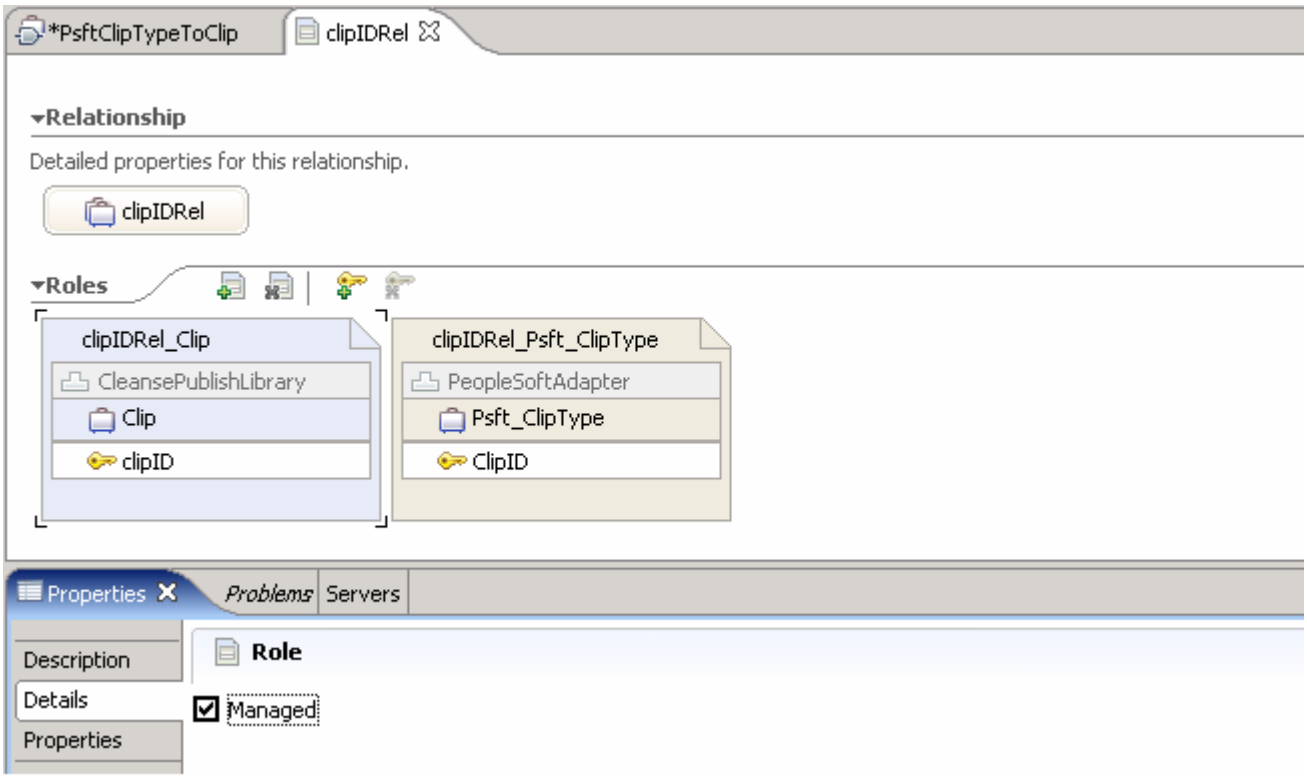
__ m. Press **Ctrl+S** to save the relationship definition.

__ n. Ensure that the **clipIDRel_Clip** box is still selected.

__ o. Select the **Properties** tab and then the **Details** property tab.

__ p. Select the **Managed** checkbox.

__ q. Press **Ctrl+S** to save the relationship definition and you can now close it.



____ 5. Add a new relationship role to the Relationship transformation in a map.

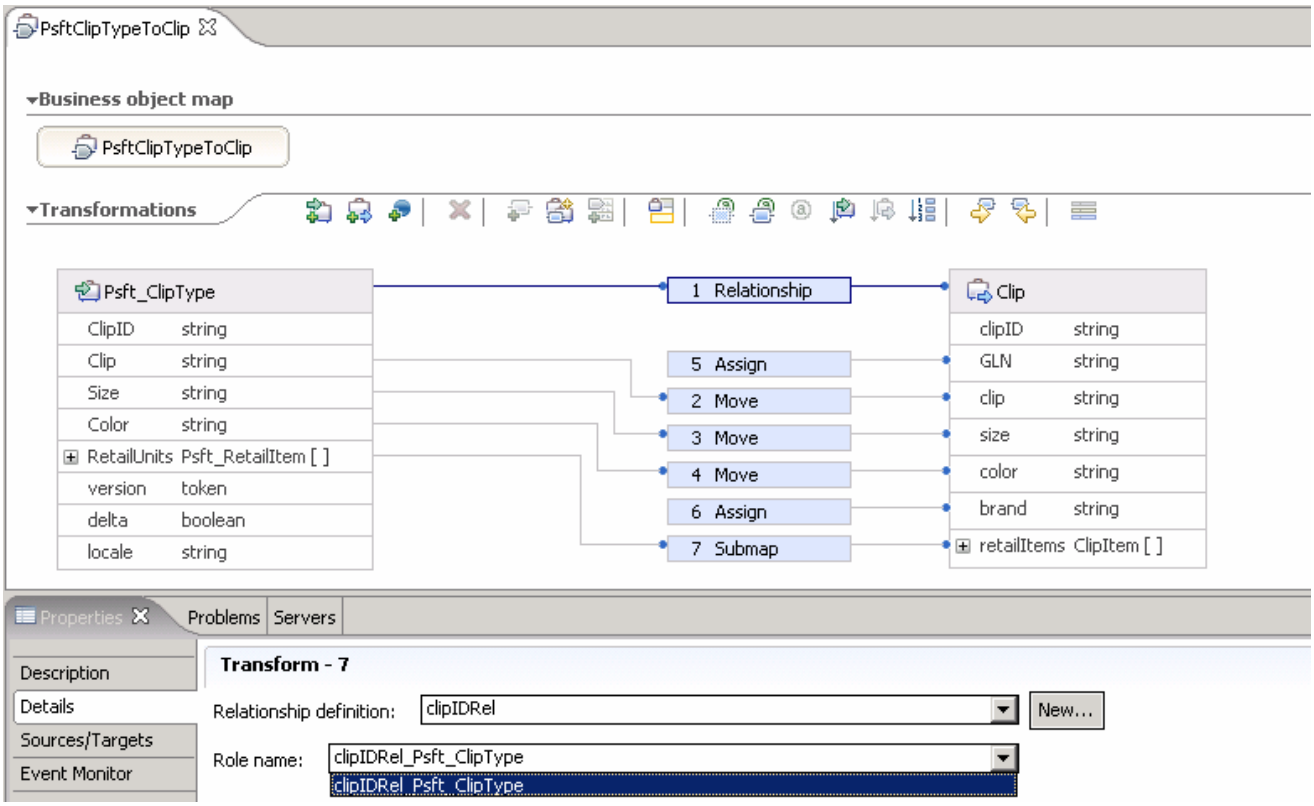
__ a. Switch to the map by clicking on the **PsftClipTypeToClip** tab in the top panel.

__ b. Select the **Relationship** transform.

__ c. Select **Properties** in the bottom panel and then the **Details** tab.

__ d. Use the dropdown box to select **clipIDRel_Psft_ClipType** for the **Role Name.**

__ e. Press **Ctrl+S** to save the map and close it.



____ 6. Create a relationship between **clipID** in **Clip** and **Key** in Siebel_ProductType.

__ a. In the Business Integration view, expand **SiebelAdapter > Mapping > Data Maps.**

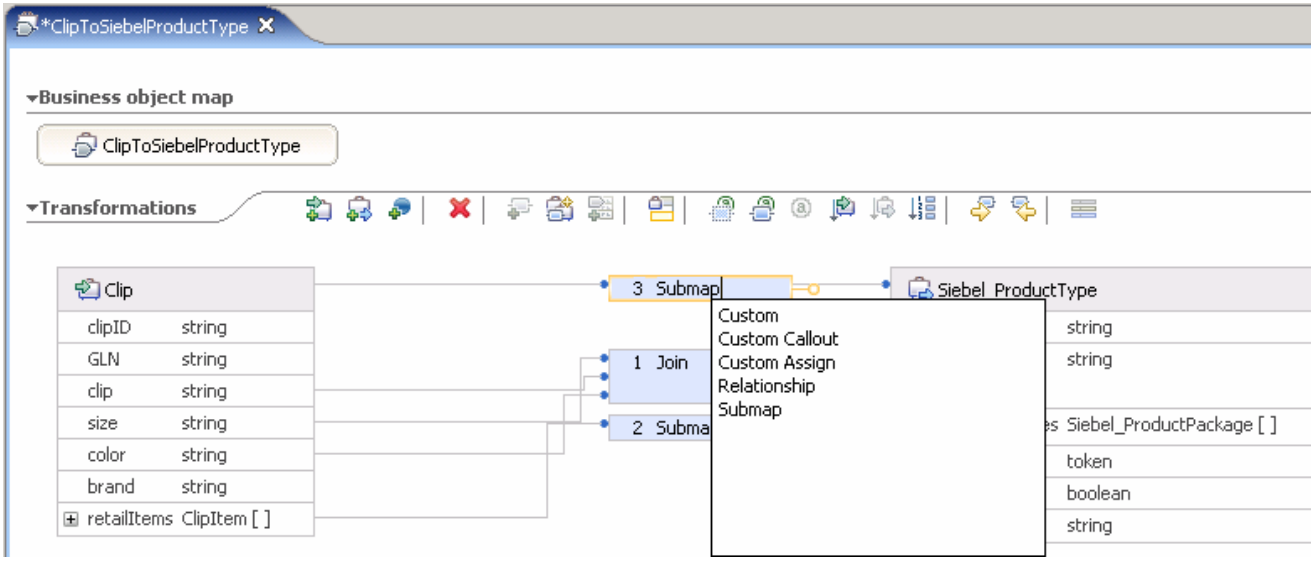
__ b. Double click on **ClipToSiebelProductType** to open it in the Mapping Editor.

__ c. Draw a line from **Clip** to **Siebel_ProductType.**

__ d. Click on the right side of the **Submap** transformation rule label.

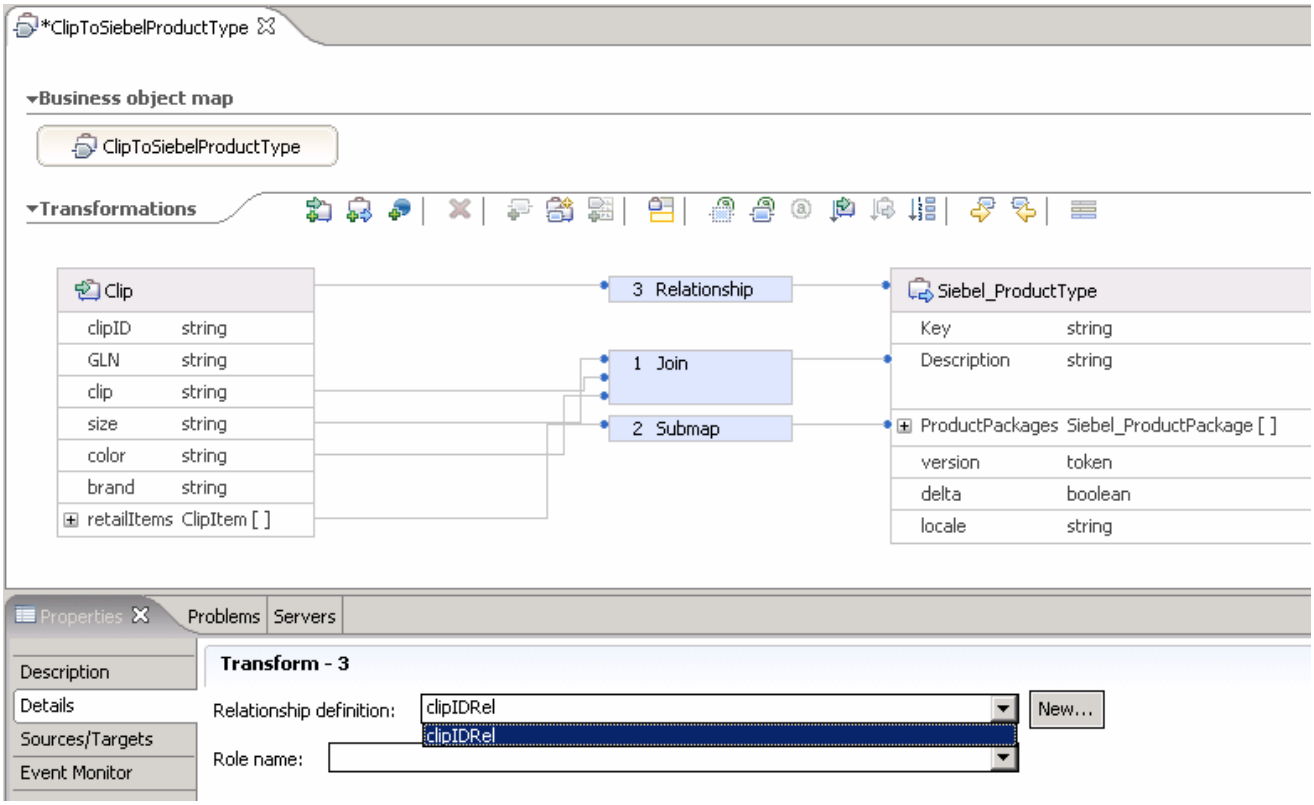
__ e. Select **Description** from the Properties view.

__ f. Change the rule from Submap to **Relationship** using the transformation dropdown.



__ g. Select the **Properties** tab at the bottom and then click on the **Details** tab.

__ h. Verify that the Relationship definition field is set correctly to: **clipIDRel.**



____ 7. Define the roles in the relationship.

__ a. Double click on **clipIDRel** to open it in the Relationship editor (if already closed).

__ b. Add a role by clicking on the **Add Role** icon.

__ c. Select **Siebel_ProductType** in the Data Type Selection dialog.

__ d. Click **OK.**

__ e. Click on the new **ClipIDRel_Siebel_ProductType** role to select it.

__ f. Notice the 4 dots at the corners indicating it is currently selected.

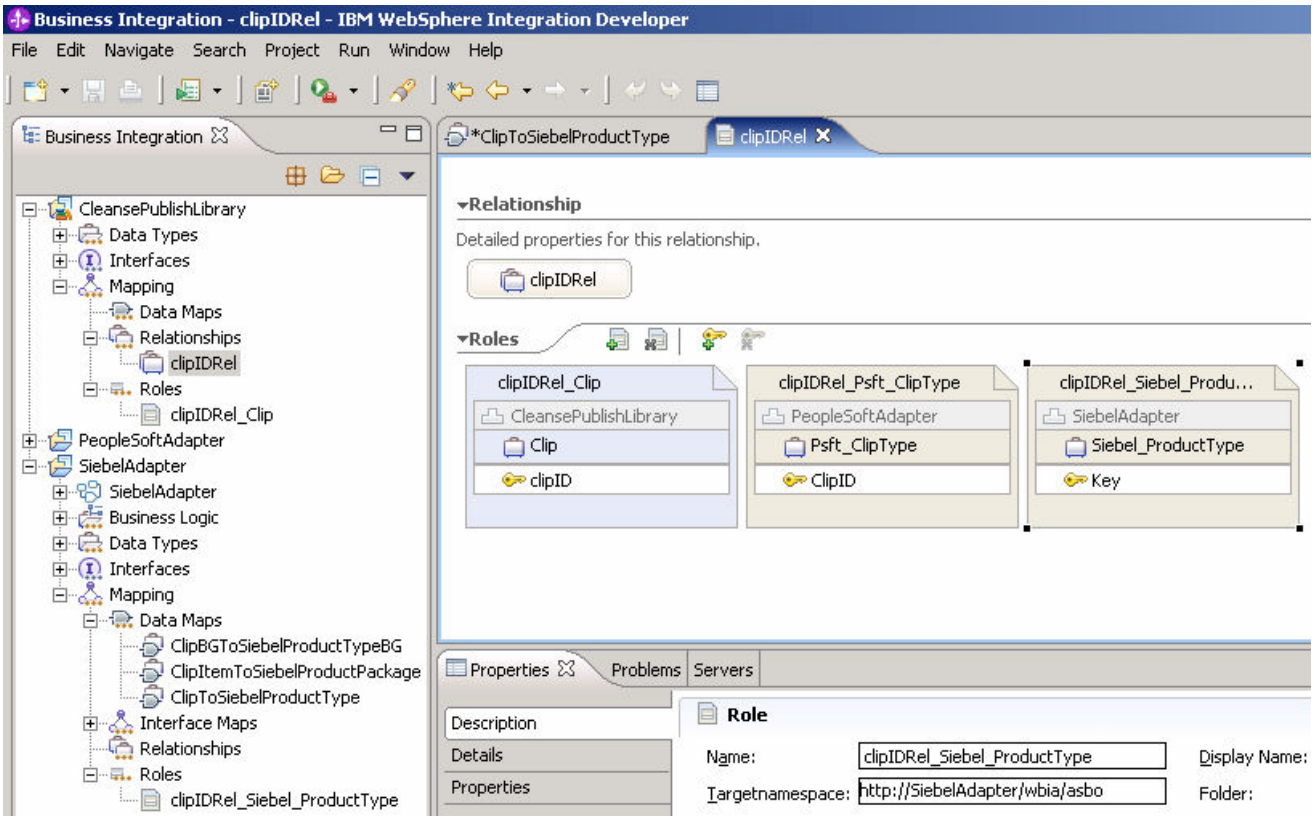
__ g. Click on the **Add Key Attribute** icon.

__ h. Select **Key** In the Select Key Attributes dialog.

__ i. Click **OK.**

__ j. Press **Ctrl+S** to save the relationship definition.

__ k. Click the **X** button on the **clipIDRel** relationship tab to close the saved relationship definition.



____ 8. Add the new relationship role to the map.

__ a. Switch to the map by clicking on the ***ClipToSiebelProductType** tab in the top panel.

__ b. Select the **Relationship** transformation rule.

__ c. Select **Description** from the Properties view.

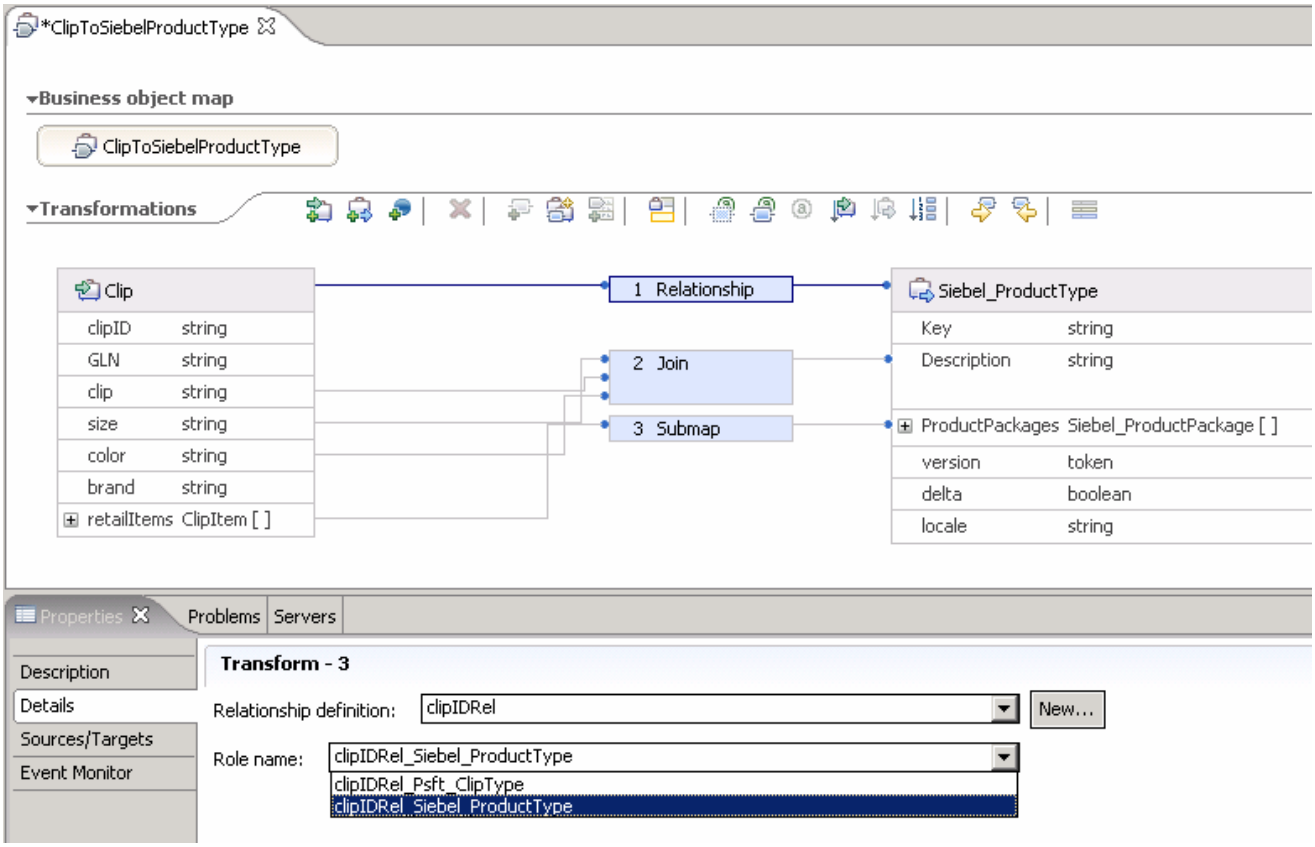
__ d. Change the **Execution order** setting to **1** for the Relationship rule from the drop down.

__ e. Select the **Properties** in the bottom panel and then the **Details** tab.

__ f. Use the drop down box to select **clipIDRel Siebel_ProductType** for the Role Name.

__ g. Press **Ctrl+S** to save the modified map.

__ h. Click the **X** button on the map tab to close the saved map definition.

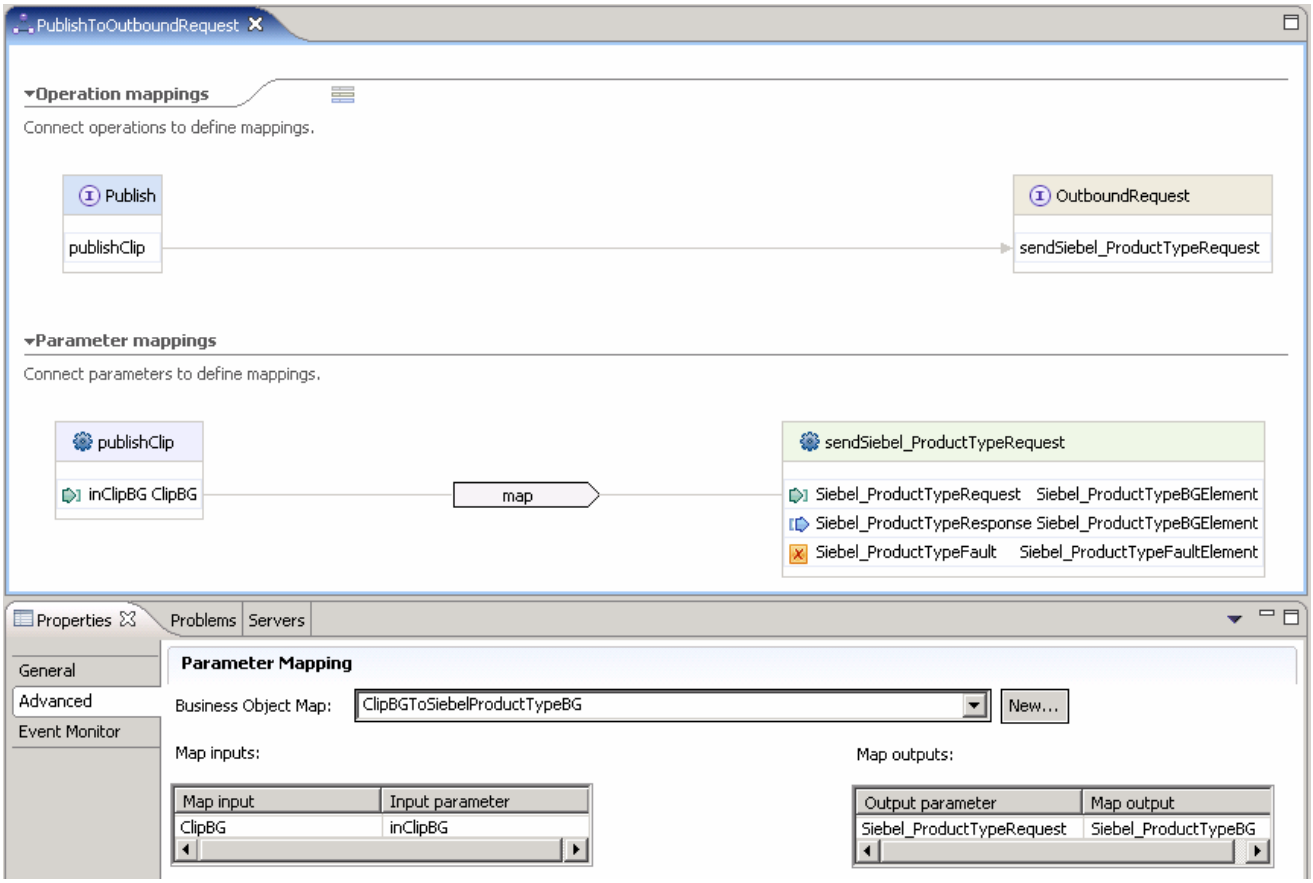


Part 3: Completing the Relationship Picture

What you did in the previous section showed you how to go through the authoring process for a Relationship. However, there is more work that is required to complete the relationship story. In this section you examine what additional work is needed without having you go through the actual authoring steps needed to create the rest of the artifacts. Then in Part 4 you will import them and run the tests.

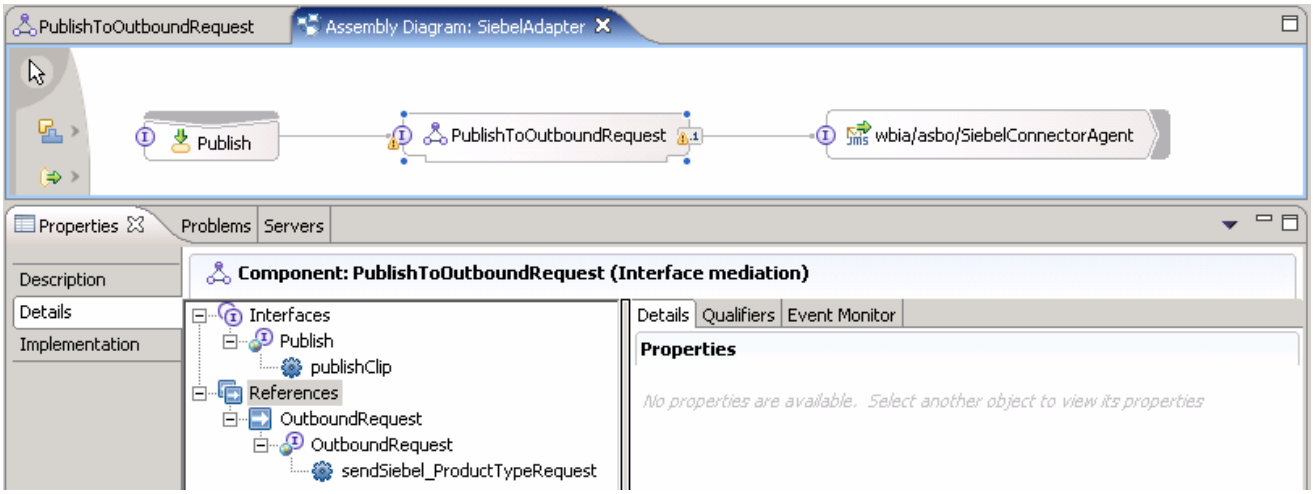
Since you do not import these until Part 4, you will not see them in WebSphere Integration Developer at this time. Simply read through Part 3 to understand what is required.

Start by looking at the **PublishToOutboundRequest** interface map that you created in the previous lab on mapping.



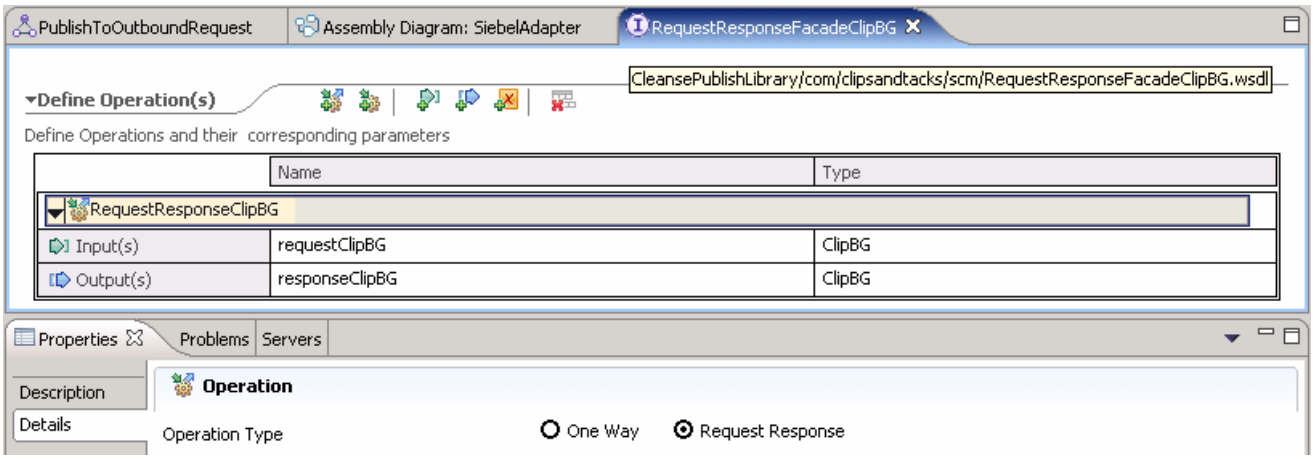
The inClipBG parameter is mapped to the **Siebel_ProductTypeRequest** parameter on the request. However, the **Siebel_ProductTypeResponse** parameter is not mapped to anything on the response. In a create scenario, there is no key associated with the **Siebel_ProductType** business object until the response comes back from Siebel. There must be a mapping operation to map the Siebel_ProductType being returned to a Clip. It is during this mapping operation that the relationship is updated with the new key value.

Probably the best way to start explaining this would be to start with the wiring diagram. This is what your wiring diagram currently looks like.

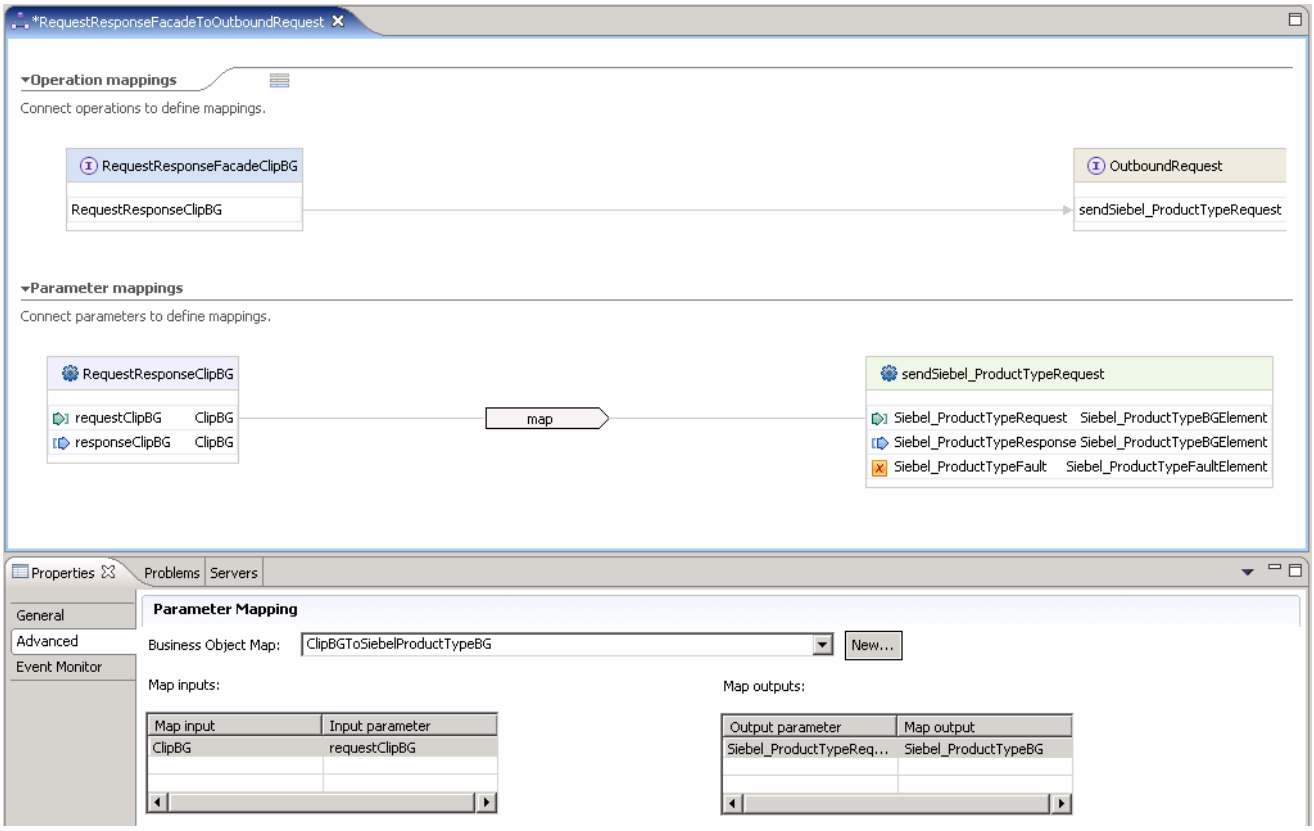


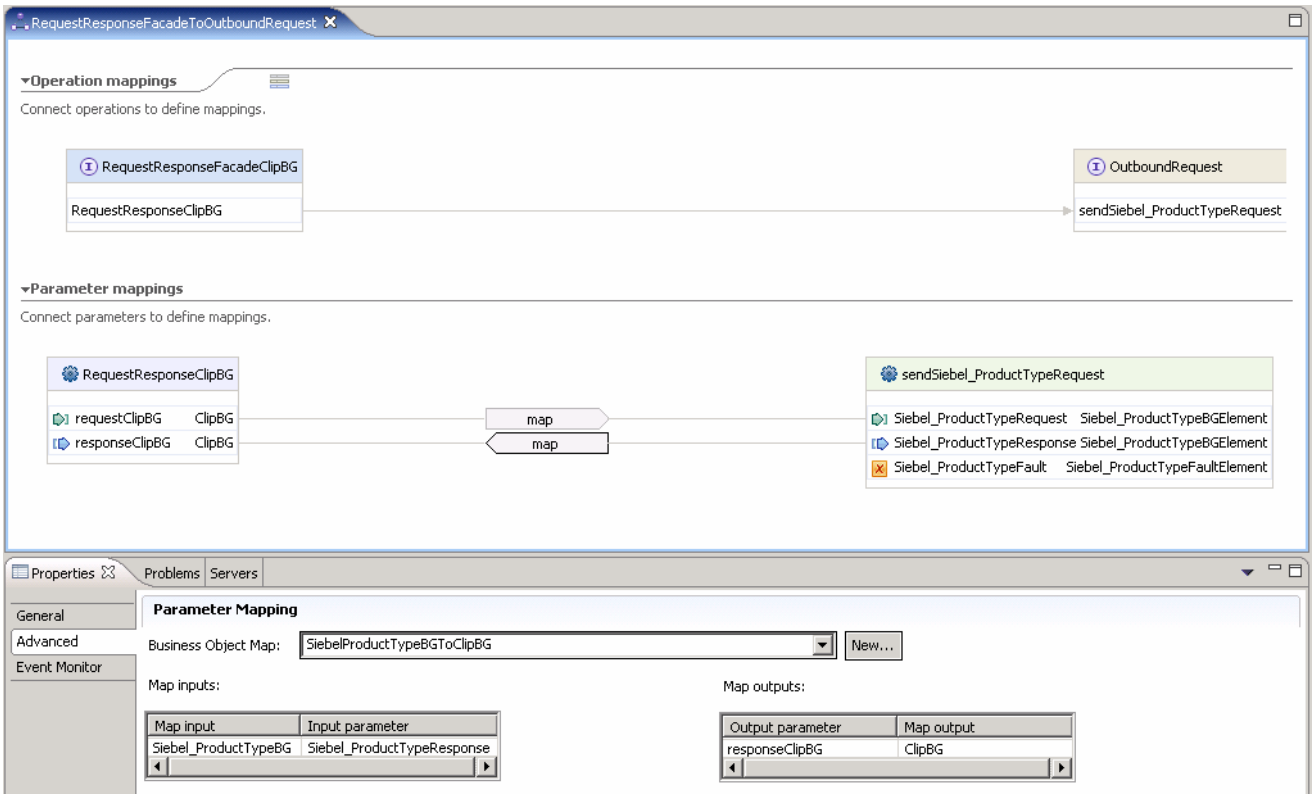
First, you are going to need three data maps to map between the Siebel ASBOs and the GBOs. In order for those maps to be invoked, the **PublishToOutboundRequest** interface map must be able to return a ClipBG. This will require a new interface on the input side. The introduction of the new interface will force another interface map to be introduced to map between the Publish interface and itself. So the list of artifacts needed would be:

- Interface
  - **RequestResponseFacadeClipBG** – has ClipBG as input & output parameters.

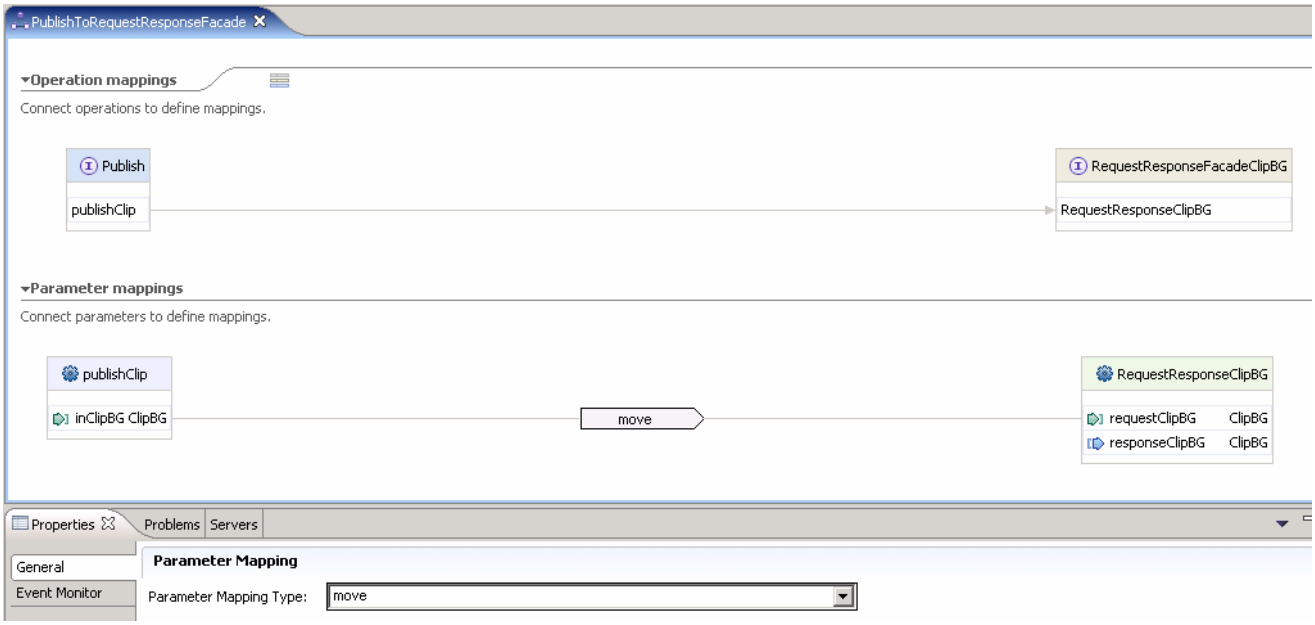


- Interface Maps
  - **RequestResponseFacadeToOutboundRequest** – replaces the existing **PublishToOutboundRequest** interface mapping.

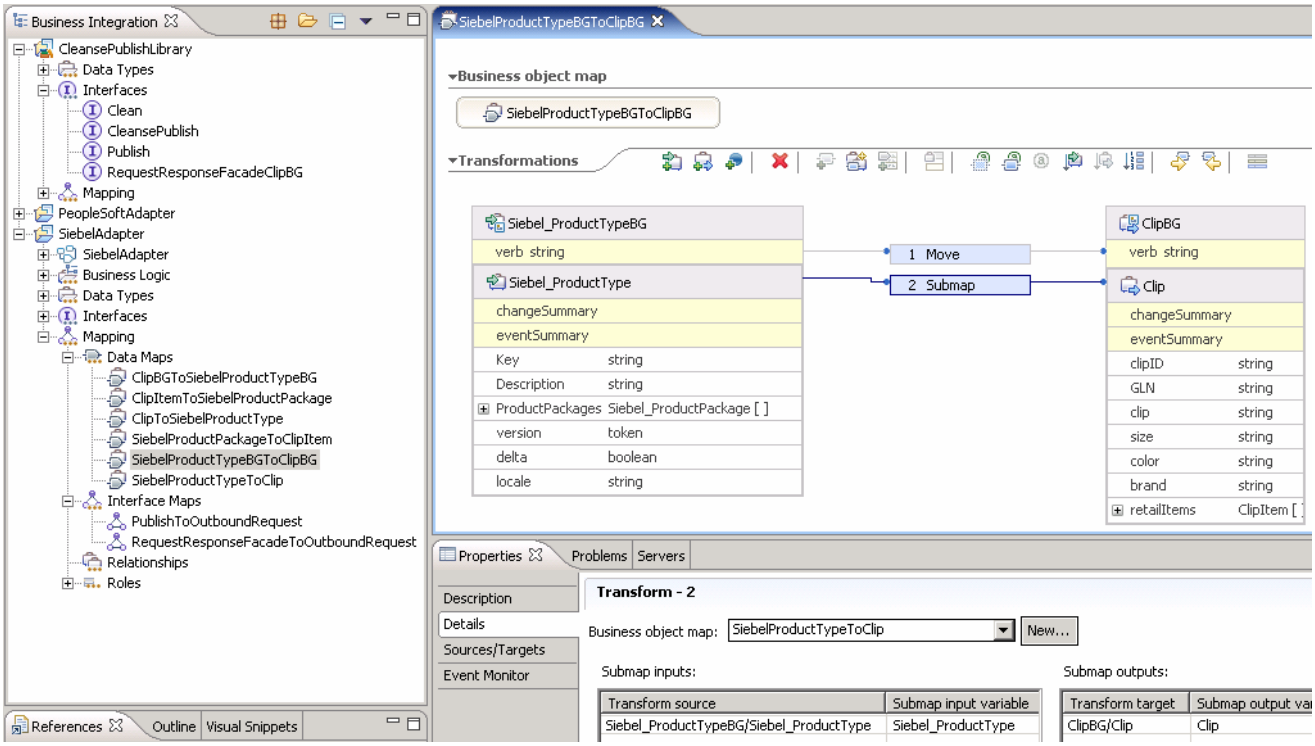




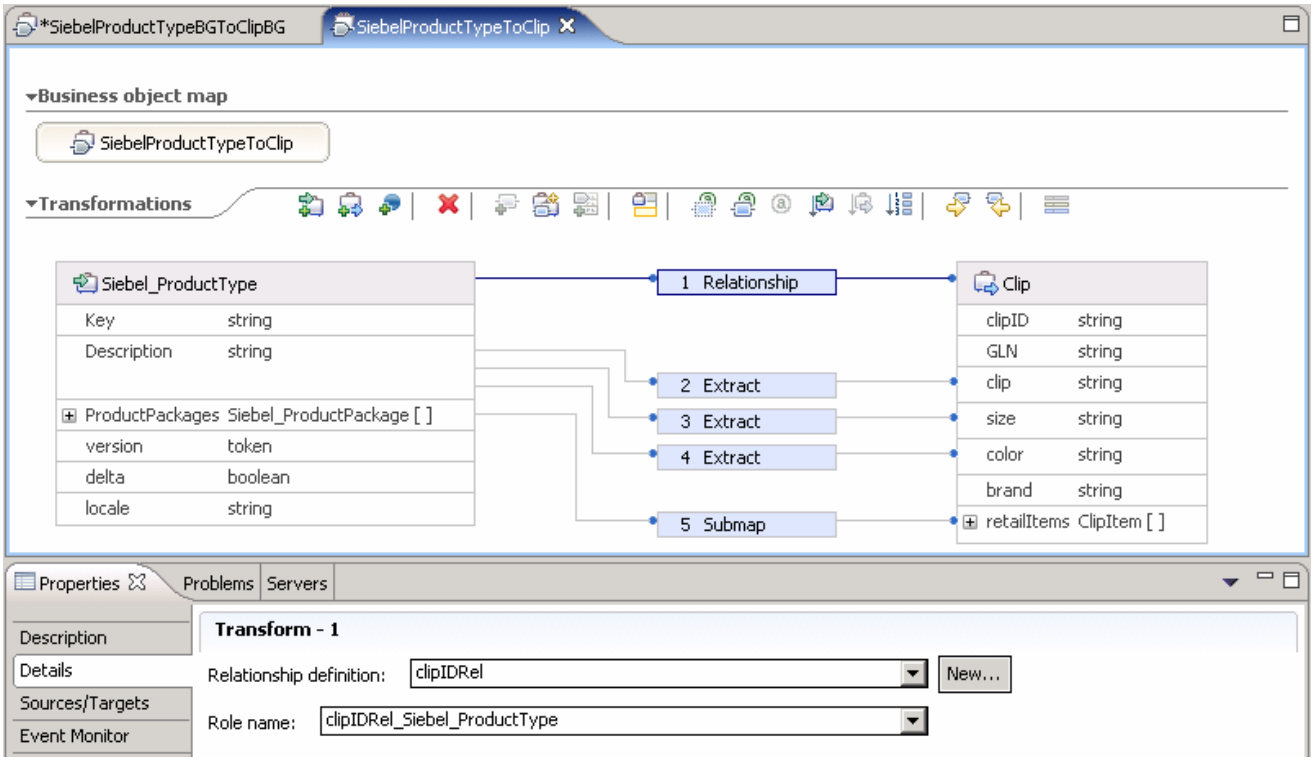
- **PublishToRequestResponseFacade** – to map from the export to the new interface map.



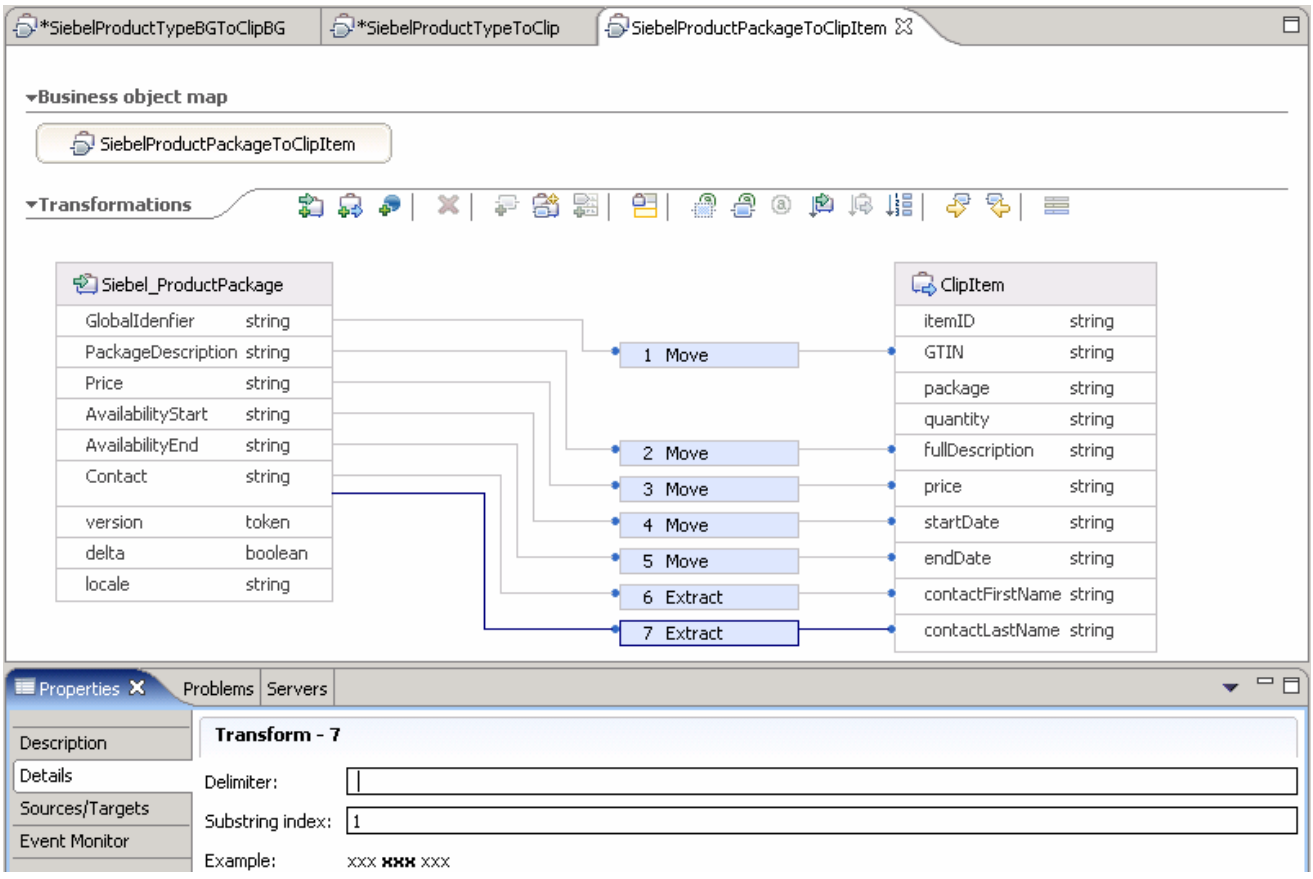
- Data Maps
  - **SiebelProductTypeBGToClipBG** – top-level map for the BG.



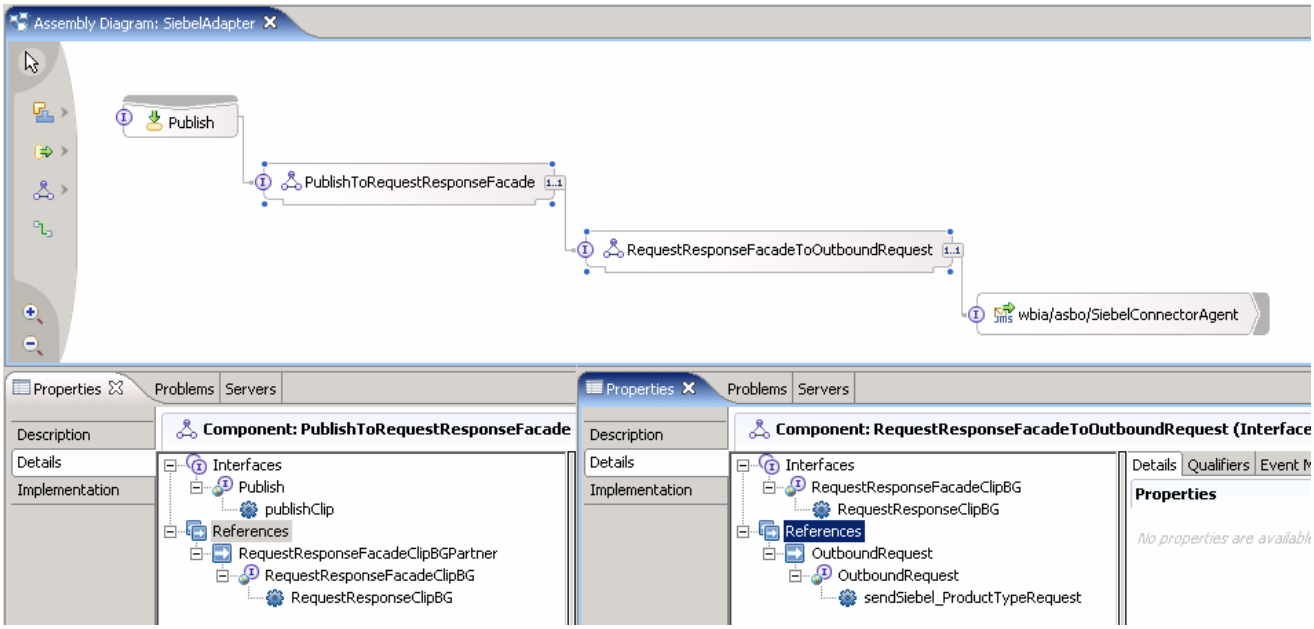
  - **SiebelProductTypeToClip** – submap for the parent BO – contains the relationship.



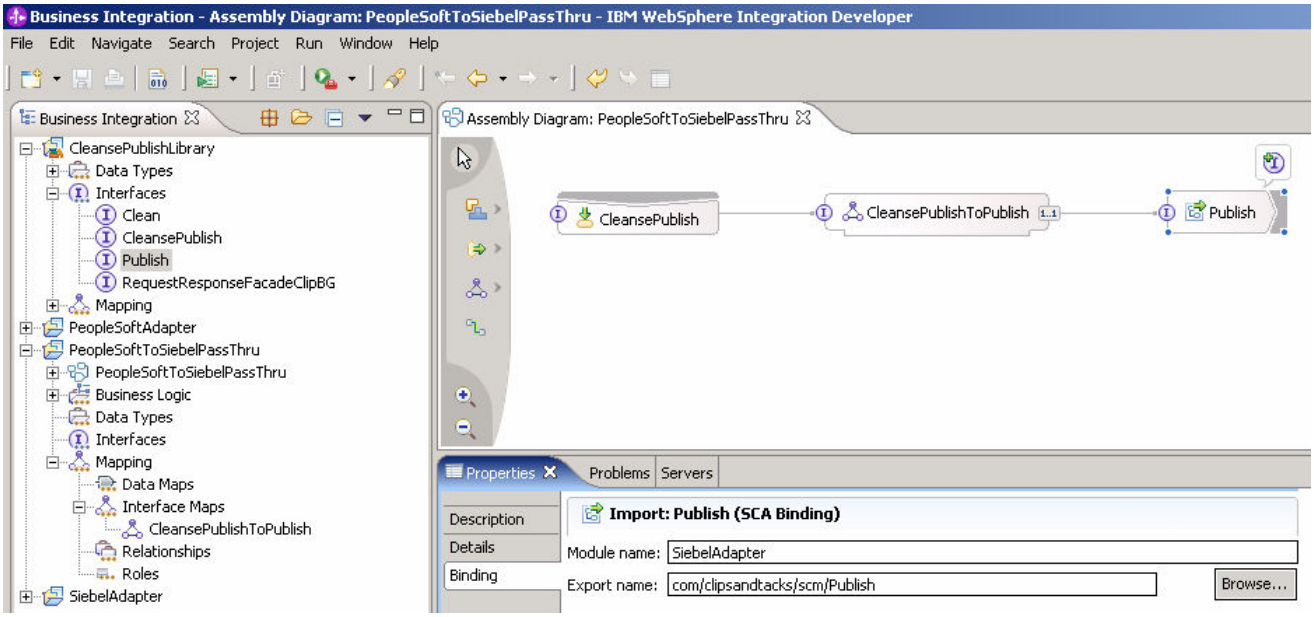
- **SiebelProductPackageToClipItem** – submap for the child BO.



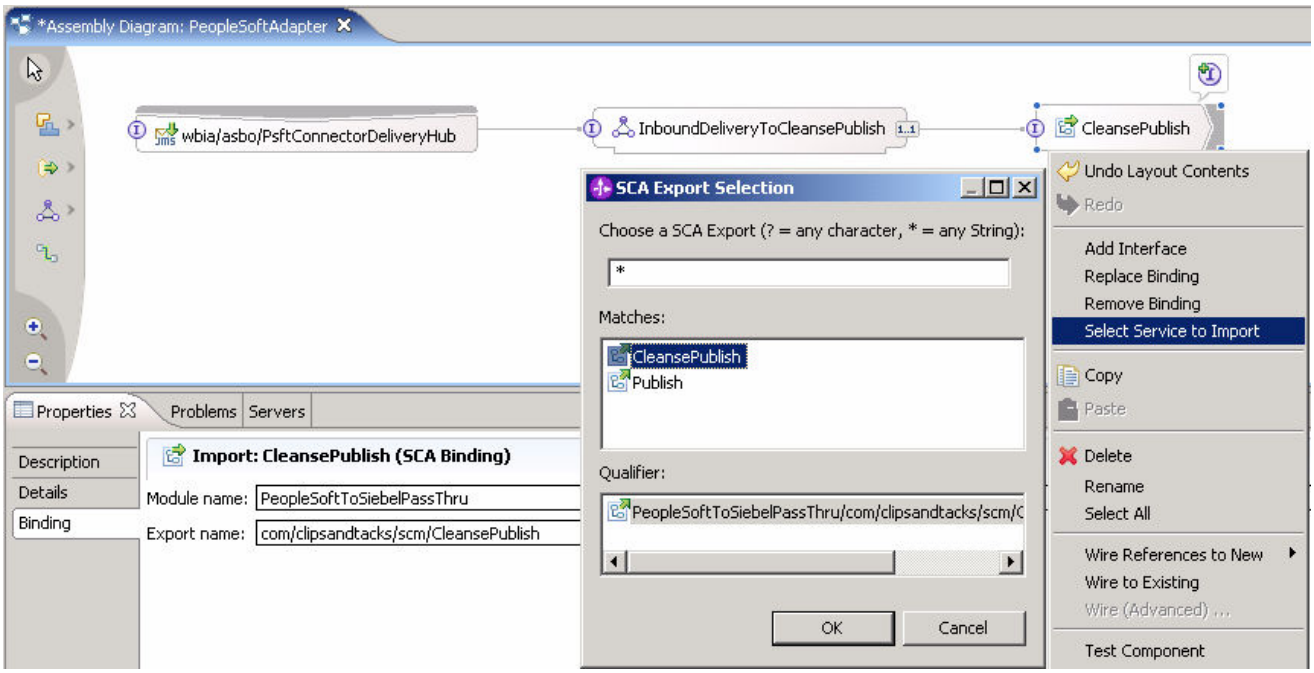
The **SiebelAdapter** module wiring diagram would now look like this:



The new **PeopleSoftToSiebelPassThru** module wiring diagram would look like this:



The **PeopleSoftAdapter** module wiring diagram would now look like this:



Part 4: Initialize the Workspace for the Testing Exercise

To complete the exercise, you would need to author everything that was described in Part 3. Rather than having you do that, a Project Interchange file has been provided which contains both of the mappings and relationships for PeopleSoft and Siebel along with the additional interfaces and mediations described in Part 3.

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

<WORKSPACE>

C:\Labfiles60\eXchange\WBIAdapters\workspaceRelsComplete

<PROJECT_INTERCHANGE>

C:\Labfiles60\eXchange\WBIAdapters\Solution\ Relationships_PI.zip

<MODULE>

n/a

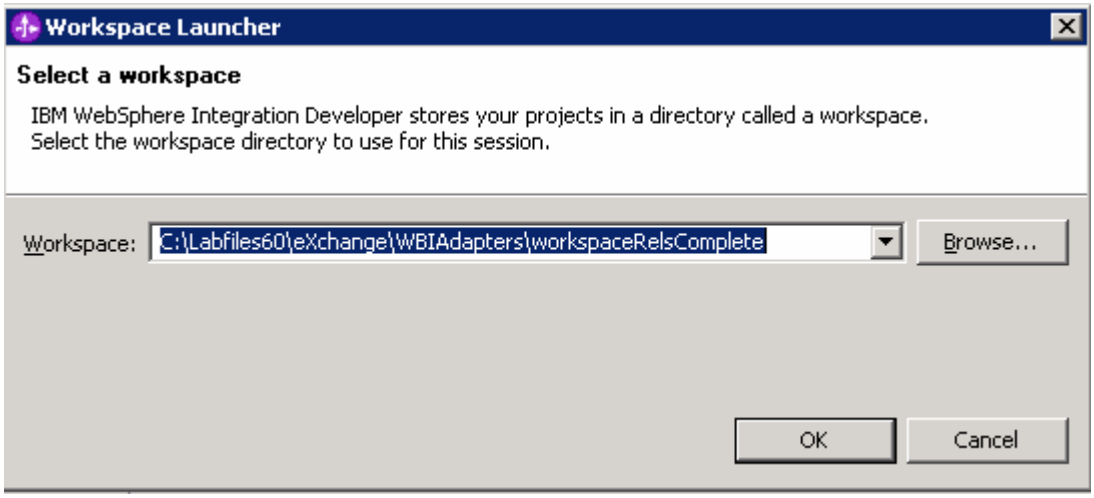
<DEPENDENT_LIBRARIES>

n/a

____ 3. 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.**



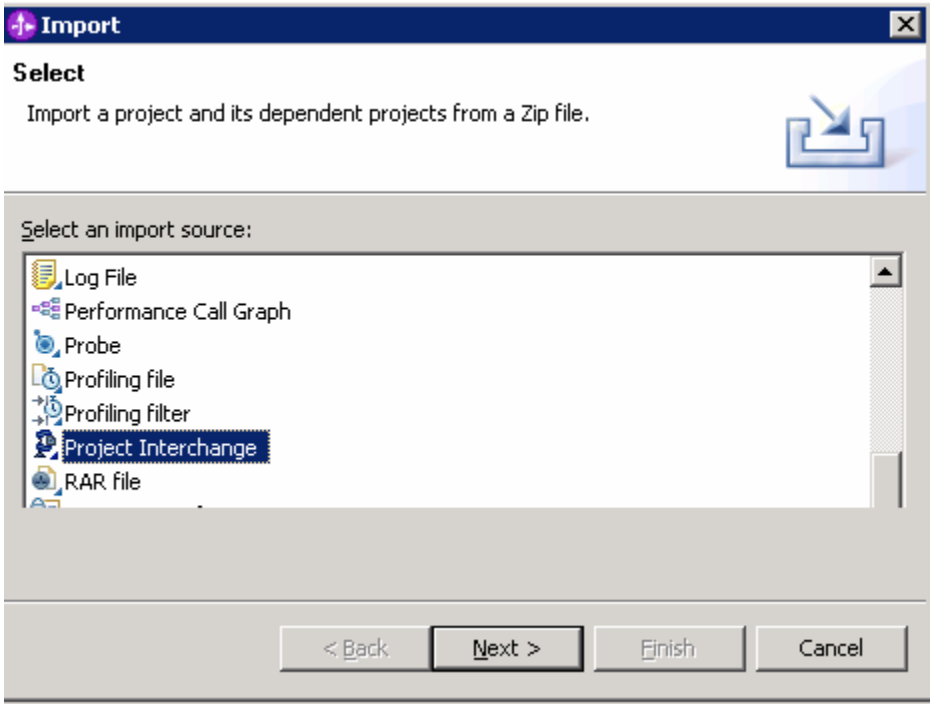
__ c. When WebSphere Integration Developer V6.0 opens, close the **Welcome page** by clicking on the Go to the workbench icon (bent over arrow at top-right).

__ d. Ensure you are in the **Business Integration** perspective.

__ e. 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>.**

__ f. From the menu bar, select **File -> Import…**

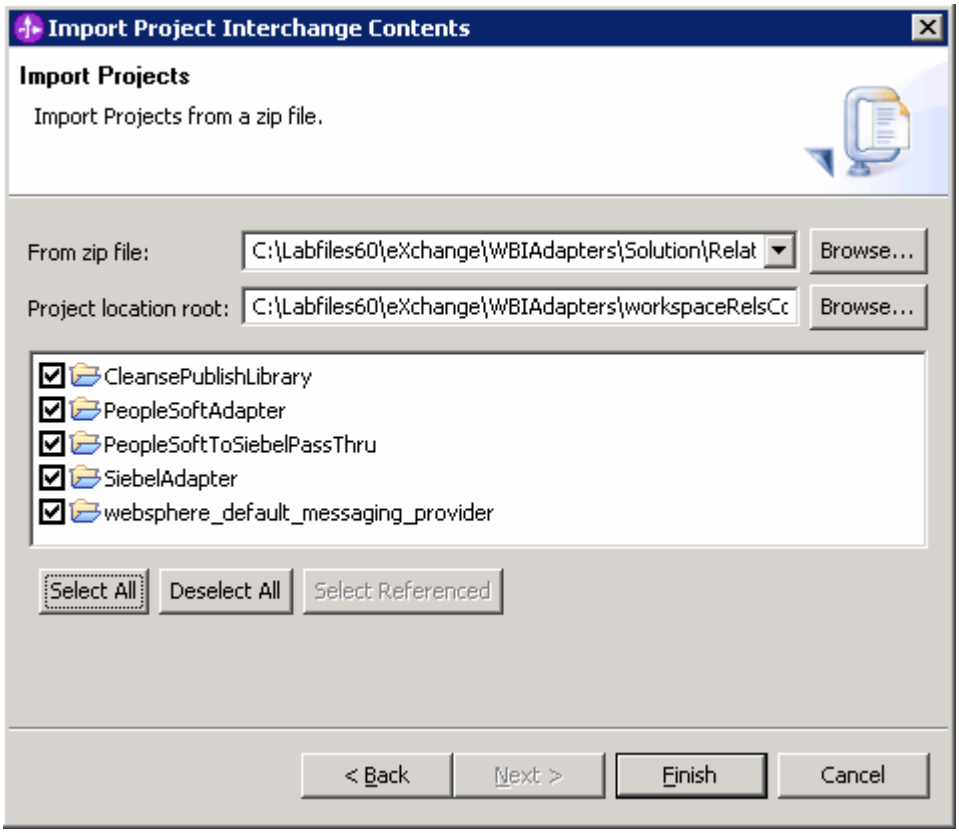
__ g. In the Import dialog, scroll down and select **Project Interchange.**



__ h. Click **Next.**

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

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



__ k. Click **Finish.**

____ 3. At this point you should be able to use the editors without instructions to examine the imported artifacts. Look at:

SiebelAdapter assembly diagram

RequestResponseFacadeClipBG Interface

PublishToRequestResponseFacade Interface Map

RequestResponseFacadeToOutboundRequest Interface Map

SiebelProductTypeBGToClipBG BG Map

SiebelProductTypeToClip BO Map (submap)

SiebelProductPackageToClipItem BO Map (submap)

____ 4. Add your projects to the server configured projects

__ a. Right click the WebSphere Process Server v6.0 server and select **Add and Remove Projects.**

__ b. If you receive a message stating that there are no projects to configure, do the following:

1) From the WebSphere Integration Developer main menu, select **Project** and disable (uncheck) **Build Automatically**

2) From the main menu, select **Project** and then select **Clean…** then **Clean All Projects**

3) From the main menu, select **Project** and then select **Build All**

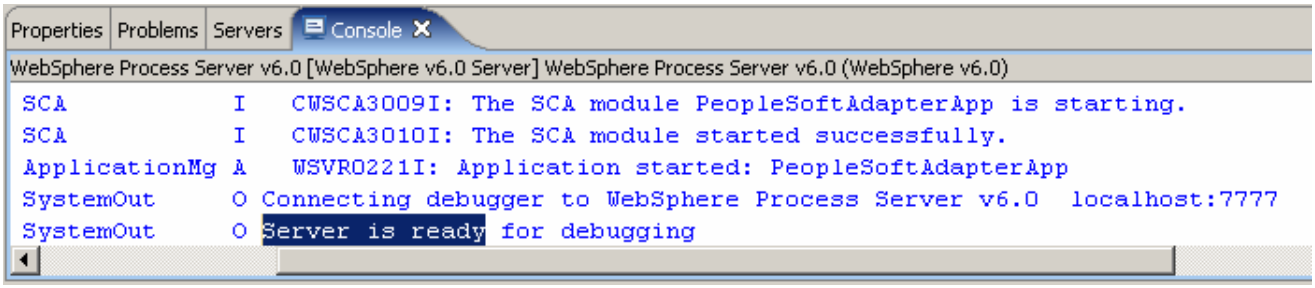
4) After the re-build, select **Project** in WebSphere Integration Developer menu and enable (check) **Build Automatically**

__ c. Click the **Add All >>** button to add the projects to the configured list.

__ d. Click **Finish.**

__ e. Wait for the server to publish the projects and to fully start (Server is ready…).

If using a remote testing environment, follow the directions provided in **Task: Adding Remote Server to WebSphere Integration Developer Test Environment** (at the end of this document) 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.



____ 5. Configure the Test Component to unit test your projects.

__ a. Right click the PeopleSoftAdapter module and select **Test > Test Module** from the menu.

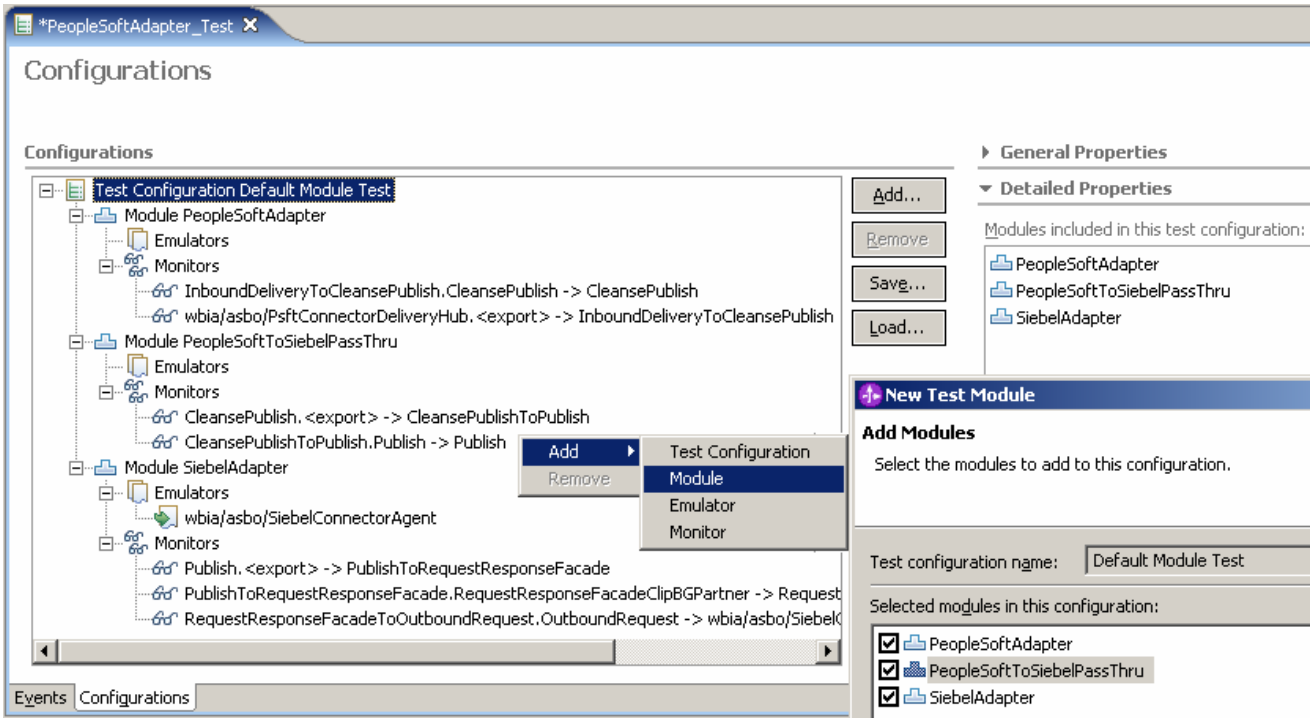
__ b. Select the **Configurations** tab at the bottom of the Test Component view.

__ c. Right click the test configuration top level node and select **Add > Module** from the menu.

__ d. Click **Next** at the initial prompt.

__ e. Click the **Select All** button to add the missing two modules to this test configuration.

__ f. Click **Finish.**



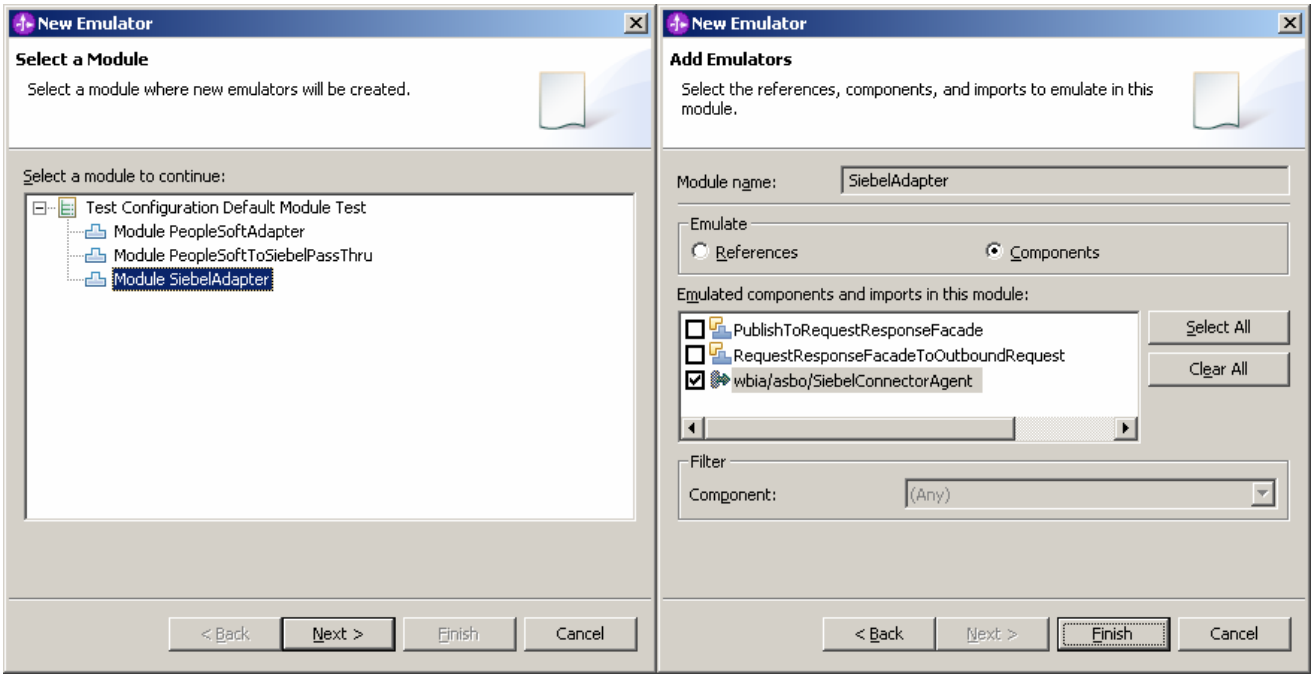
____ 6. Right click the **Module SiebelAdapter** node and select **Add > Emulator** from the menu.

__ a. Click **Next** at the initial prompt.

__ b. Change the **Emulate** radio button option to **Components.**

__ c. Enable (select) the **wbia/asbo/SiebelConnectorAgent** component.

__ d. Click **Finish.**



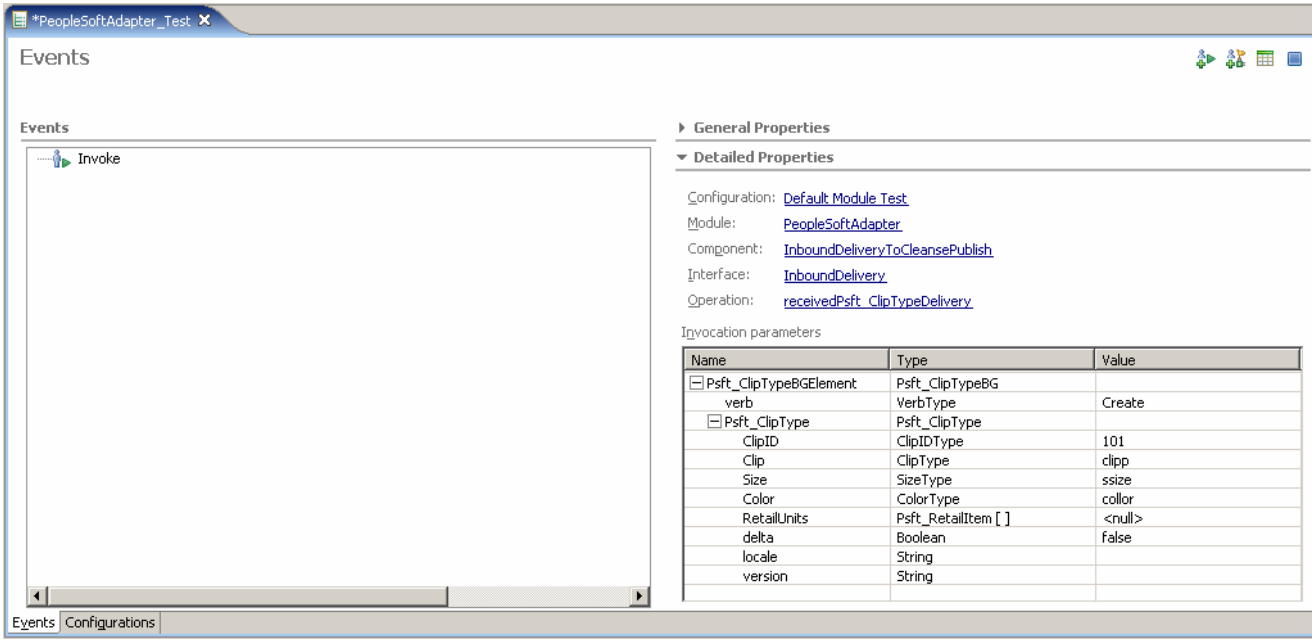
____ 7. Invoke the Test Component and unit test your projects.

__ a. Select the **Events** tab at the bottom of the Test Component view.

__ b. Make sure that the **Detailed Properties** match with the following screen capture.

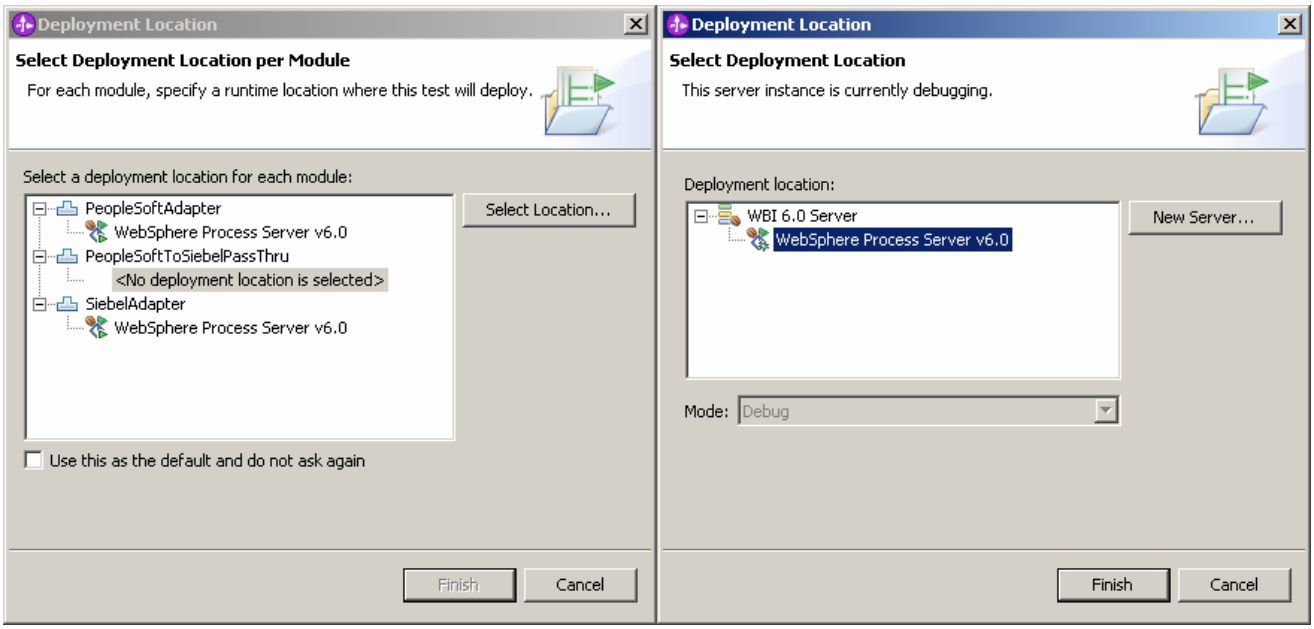
__ c. You might have to select the **Component** under Detailed Properties as **InboundDeliveryToCleansePublish** using the drop down menu.

__ d. Provide sample data for the Psft_ClipTypeBGElement, you must provide the **verb** (such as Create) and a unique **ClipID** value.



__ e. Click the **Continue** button beneath the Invocation parameters view.

__ f. Select each module that does not have the deployment location and then click **Select Location.**

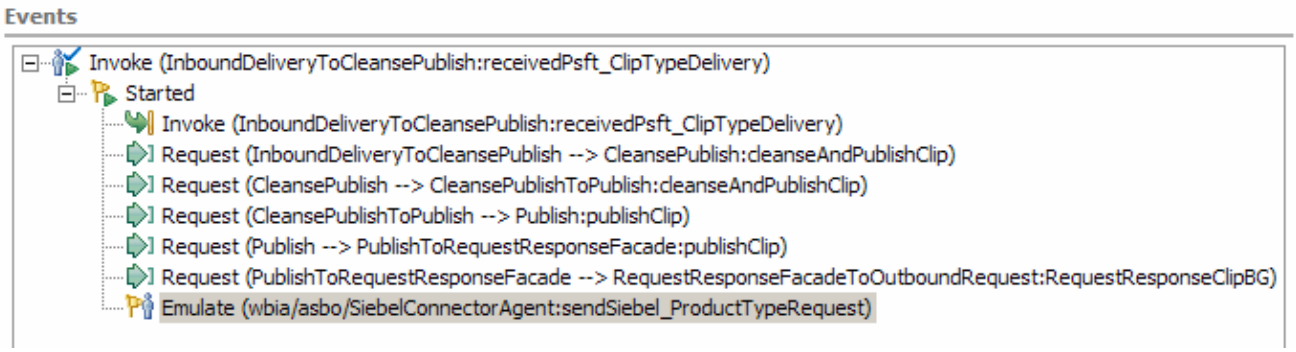


__ g. Select the appropriate WebSphere Process Server on which you want to deploy the application. This is especially true for z/OS, AIX, Solaris remote test environment, where the WebSphere Integration Developer will be remote to the test environment. Click **Finish**

Note: If you get exceptions in the test component window, stop and close the test component and then restart the server with the applications still added to the server. When the server finishes restarting, continue from step 4 in part 4. If using a remote testing environment, follow the directions provided in **Task: Adding Remote Server to WebSphere Integration Developer Test Environment** (at the end of this document) to restart a server.

____ 8. Observe the Test Component start routine and wait for it to reach the Emulator event.

__ a. Examine the already run events once the Test Component pauses.



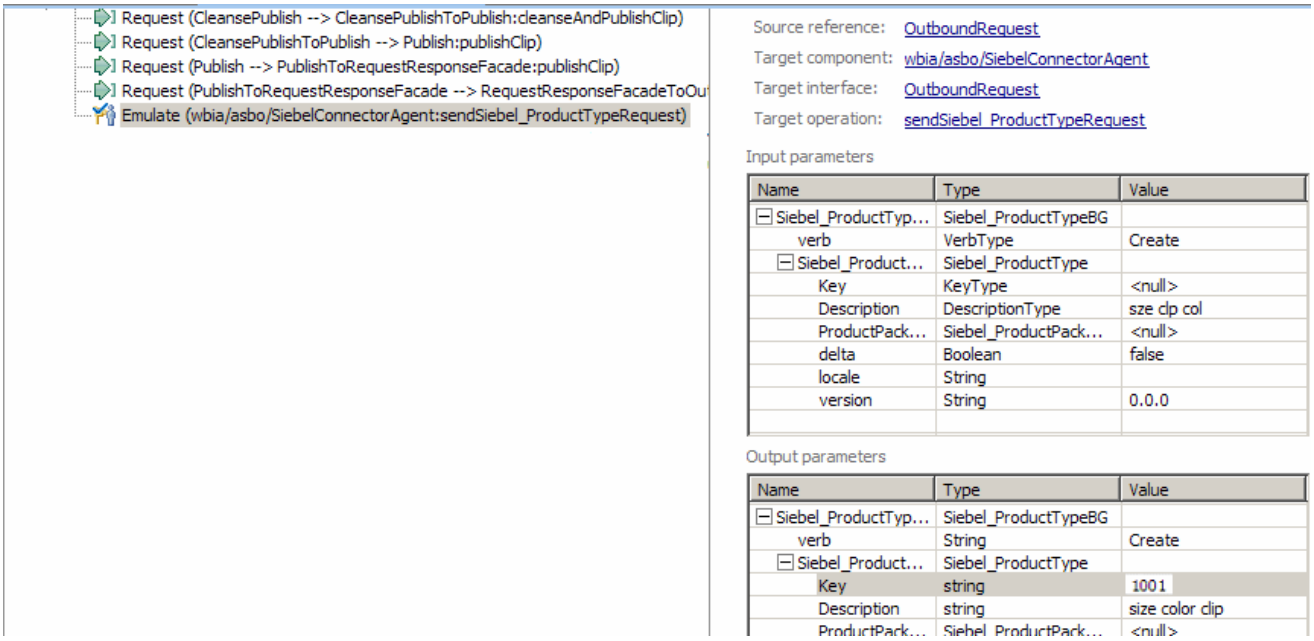
__ b. Select the Emulate (wbia/asbo/SiebelConnectorAgent:sendSiebel_ProductTypeRequest) event.

__ c. Scroll down in the **Detailed Properties** view and locate the **Output parameter** view.

__ d. You must provide the response data emulating the Siebel Adapter responding to the system. This can be done by doing a copy of **Siebel_ProductTypeBGElement** in the Input Parameters and then doing a paste to **Siebel_ProductTypeBGElement** in the Output Parameters.

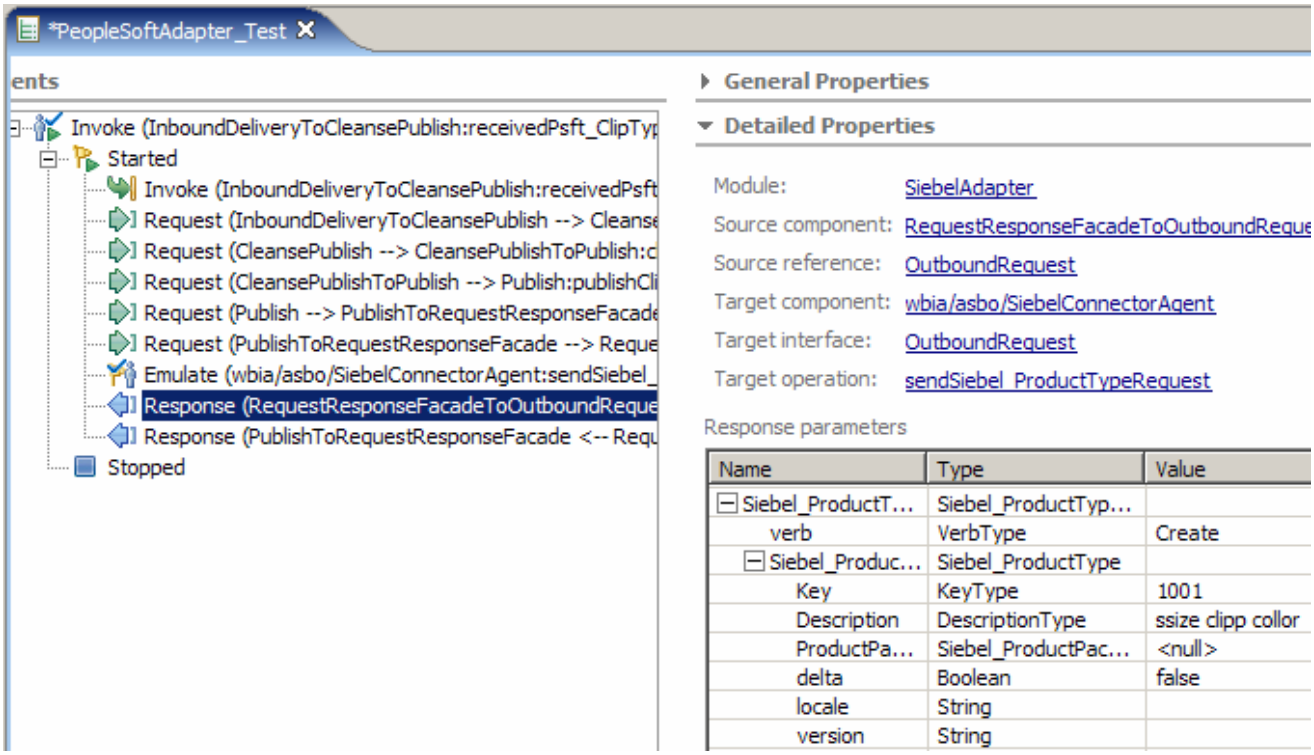
__ e. Ensure you entered the correct **verb** when starting the Test Component (Create).

__ f. Enter a unique Siebel **key** value (emulating the Siebel application creating a new entry in its data store). For example, 1001 as shown in the screen capture.

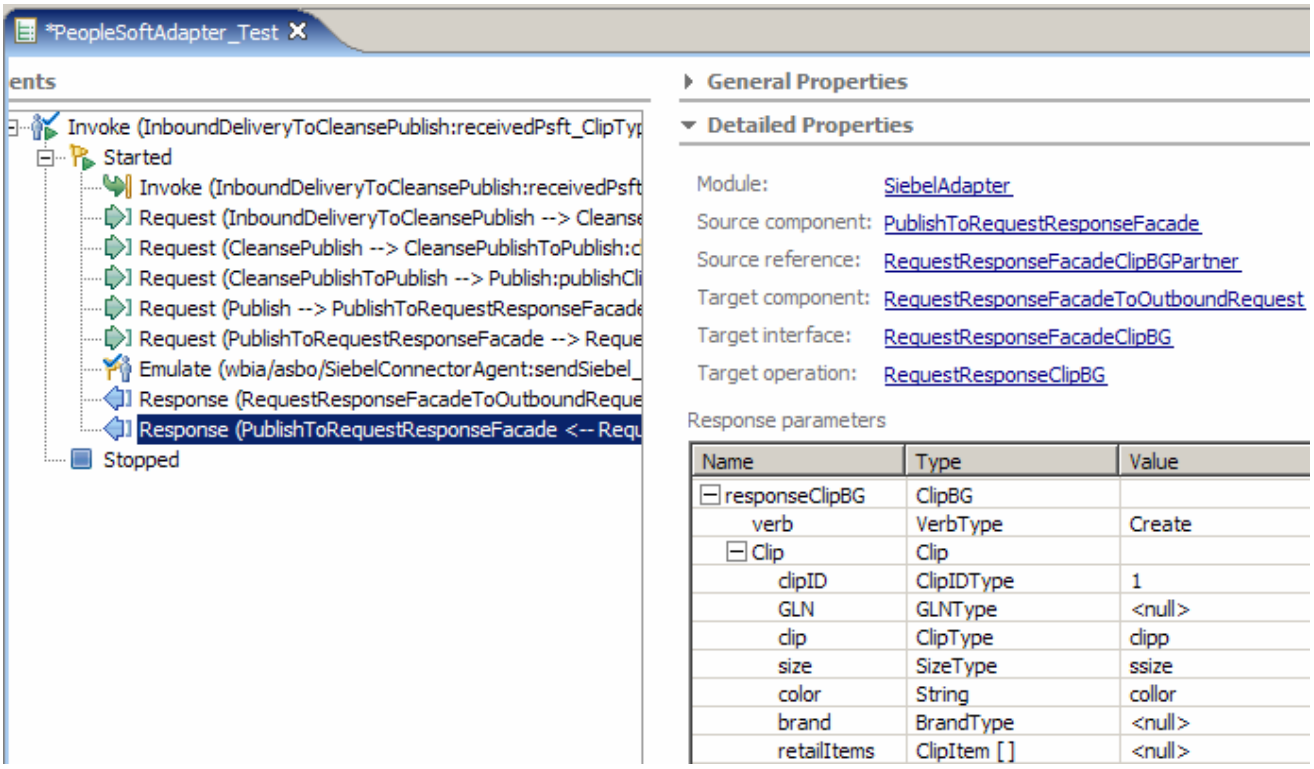


__ g. Click the **Continue** button and observe the Test Component finishing up the unit test flow.

__ h. Select the first **Response** event and verify the response Siebel data sent back to the system.



__ i. Select the second **Response** event and verify that the response ClipBG object unique key (clipID) matches the unique key value that assigned to the request ClipBG by the RelationshipService.



____ 9. Rerun the Invocation from within the already configured Test Component several times, using unique values for both PeopleSoft and Siebel BG primary keys each time.

____ 10. Right click the server and **Remove all** projects from your server instance.

____ 11. Stop the server before proceeding with the next exercise.

__ a. Right click on WebSphere Process Server v6.0 server from the Servers view and select Stop from the context menu.

____ 12. Examine the Relationship data stored in the WPSDB database (in Cloudscape).

__ a. In Windows Explorer, navigate to <WID_HOME>\runtimes\bi_v6\cloudscape\bin\embedded.

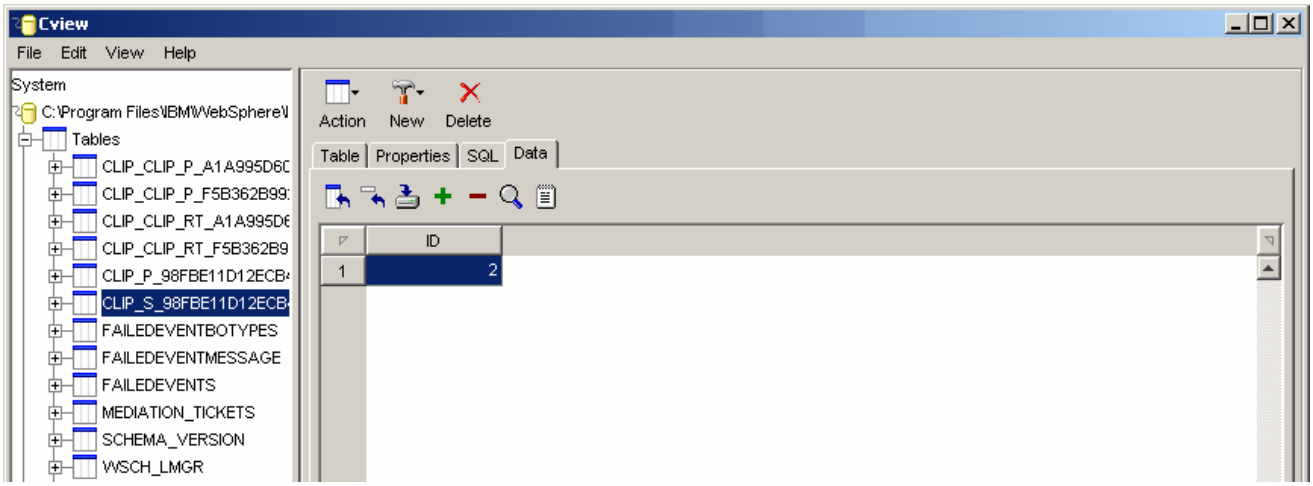
__ b. Double-click the **cview.bat** file to start the **Cloudview** utility to manage Cloudscape databases.

__ c. Select **File > Open…** and navigate to <WID_HOME>\runtimes\bi_v6\cloudscape\databases.

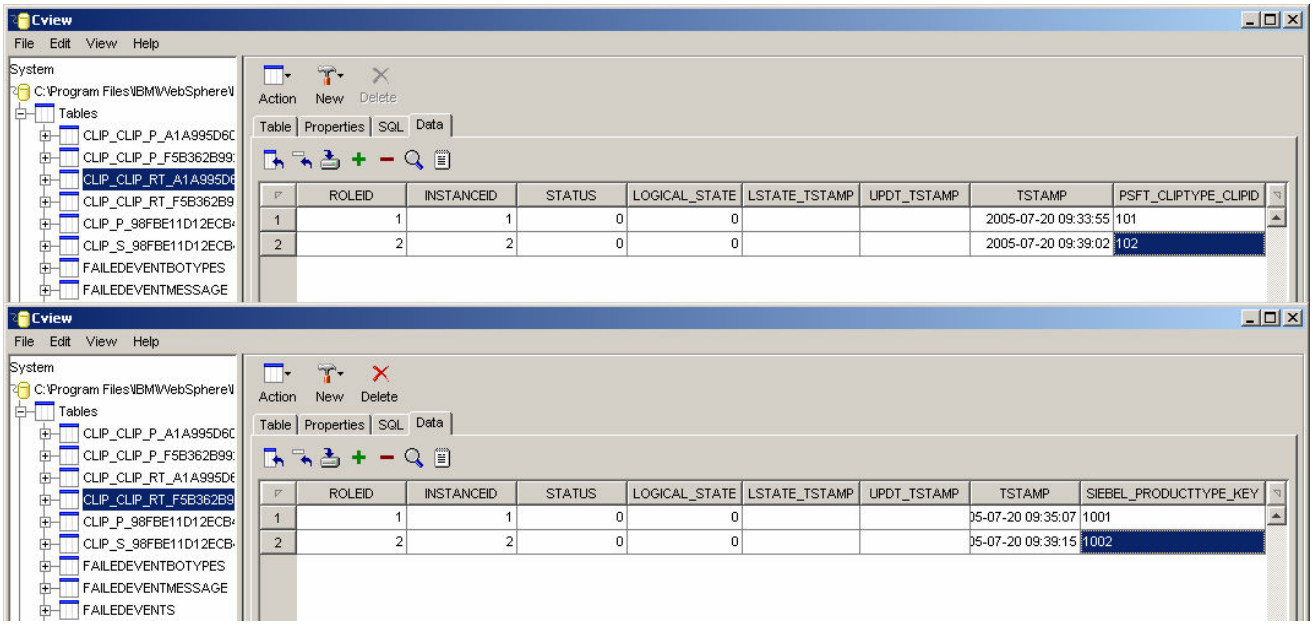
__ d. Select the **WPRCSDB** database and click **Open.**

__ e. Expand the **Tables** node and examine the list of existing database tables.

__ f. Select the **CLIP_S_*** ** table and then select the **Data** tab in the right view pane. This table stores the sequence counter for the managed clipID relationship role.



__ a. Select the **CLIP_CLIP_RT_*** ** tables and examine their content. These tables store the relationship information for the non-managed relationship roles.



____ 13. Select **File > Exit** to close the Cloudview utility when done examining the relationship data.

What you did in this exercise

You learned how to author identity relationships and how they are associated with data maps. You also learned about the requirements for an identity relationship to be completed during a create flow. Because the original design did not account for this, the design changes required were imported.

Solution Instructions

There are no solution instructions for this lab exercise.

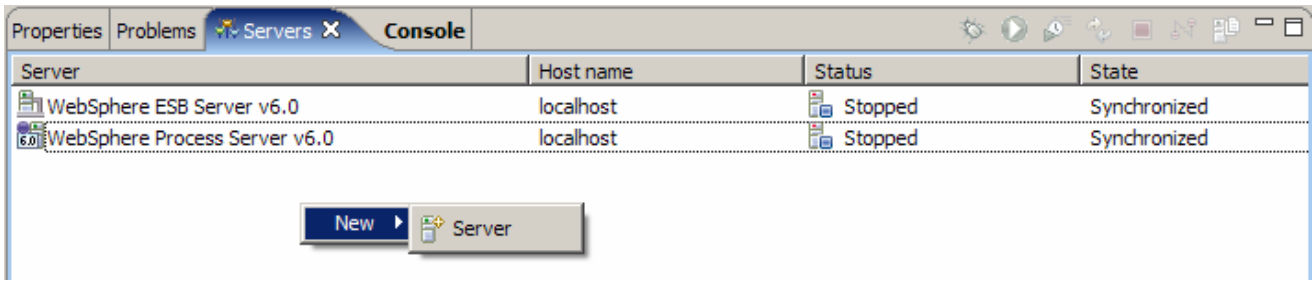
Task: Adding Remote Server to WebSphere Integration Developer Test Environment

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.

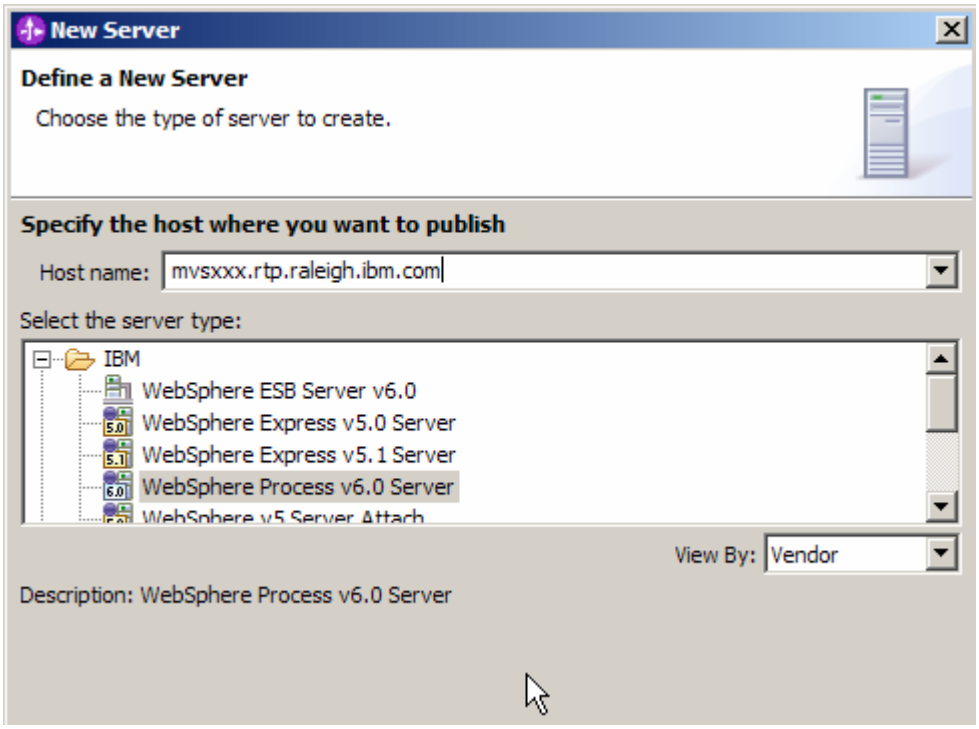
__ a. Right click on the background of the Servers view to access the pop-up menu.

__ b. Select **New > Server.**



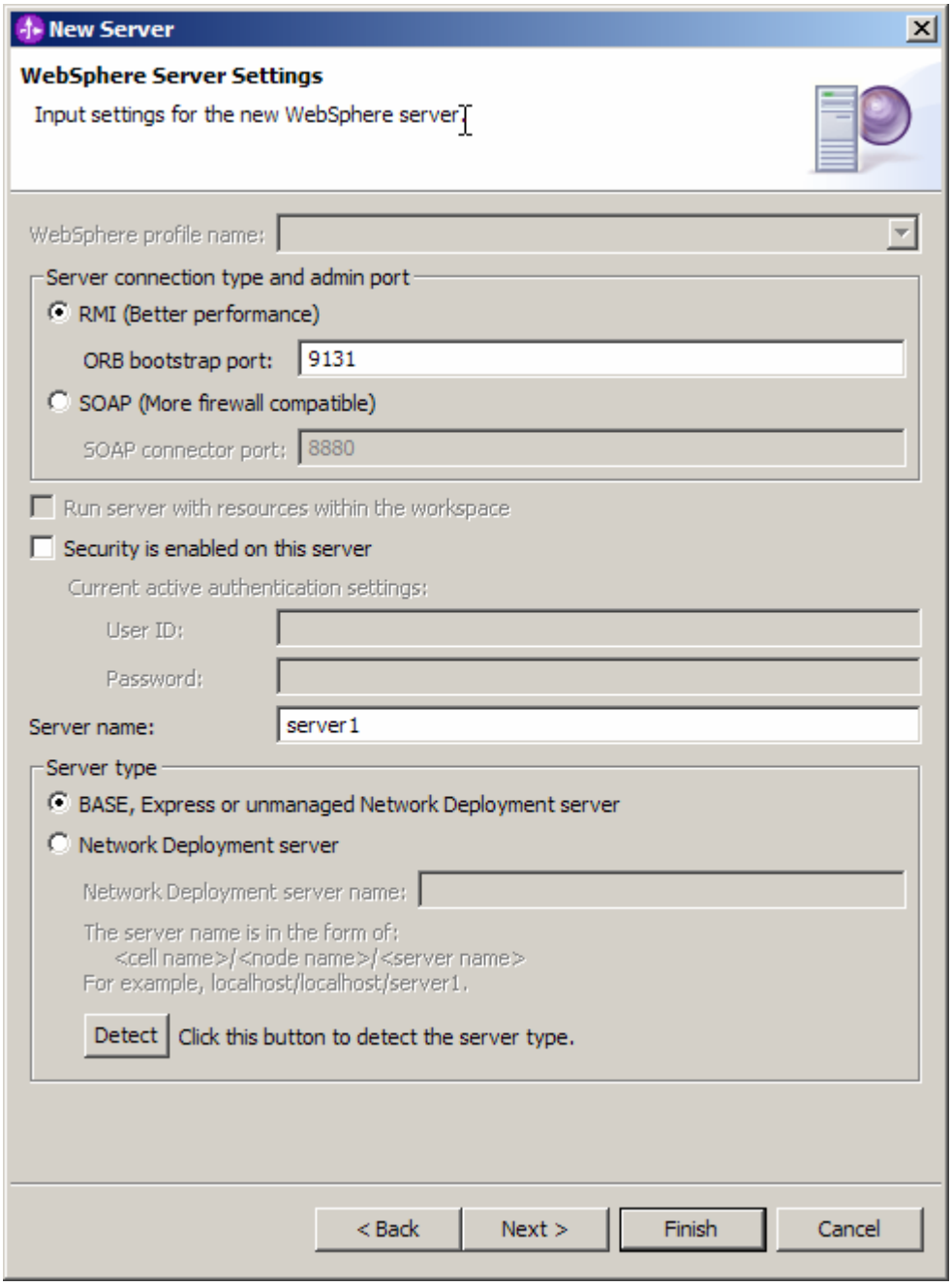
__ c. Specify hostname to the remote server, **<HOSTNAME>**.

__ d. Ensure that '**WebSphere Process v6.0 Server**' is highlighted in the server type list.



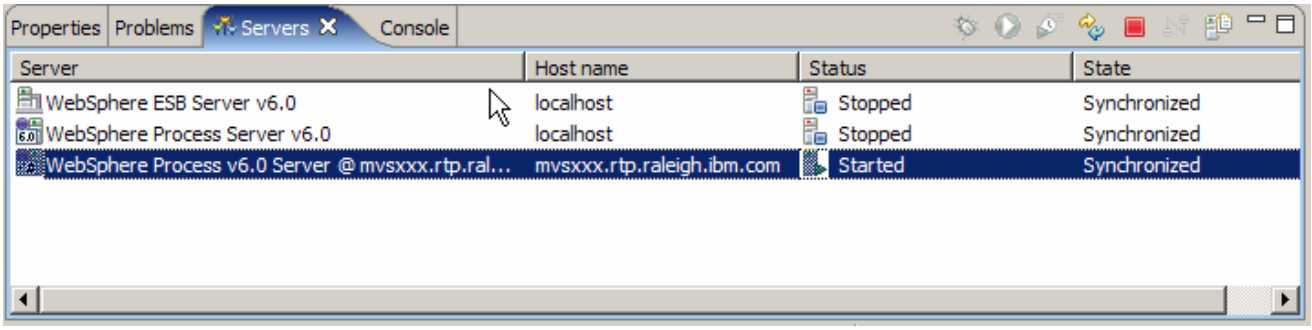
__ e. Click **Next.**

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



__ g. Click **Finish**.

__ h. The new server should be seen in the Server view.



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

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

'**telnet <HOSTNAME> <TELNET_PORT>**'

userid : **<USERID>**

pw : **<PASSWORD>**

__ j. Navigate to the bin directory for the profile being used:

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

__ k. Run the command file to start the server: **./startServer.sh <SERVER_NAME>**

__ l. Wait for status message indicating server has started:

```
ADMU3200I: Server launched. Waiting for initialization status.

ADMU3000I: Server cl1sr01 open for e-business; process id is 0000012000000002
```