

## Mapping Between ASBOs and GBOs

What this exercise is about .....	1
Lab Requirements .....	1
What you should be able to do .....	1
Introduction .....	2
Exercise Instructions .....	3
Part 1: Initialize the Workspace for this Lab Exercise.....	5
Part 2: Define Interface Maps and Data Maps for PeopleSoft.....	12
Part 3: Define Interface maps and Data maps for Siebel .....	29
Part 4: Test the PeopleSoft Maps .....	37
Part 5: Test the Siebel Maps.....	47
What you did in this exercise .....	56
Solution Instructions.....	57
Task: Adding Remote Server to WebSphere Integration Developer Test Environment.....	58

### What this exercise is about

This exercise shows you how to develop Interface maps (mediations) and Data maps. These are used to map between the PeopleSoft ASBOs and the GBOs and from the GBOs to the Siebel ASBOs.

### Lab Requirements

List of system, software and other tasks 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
- Visual Test Connector

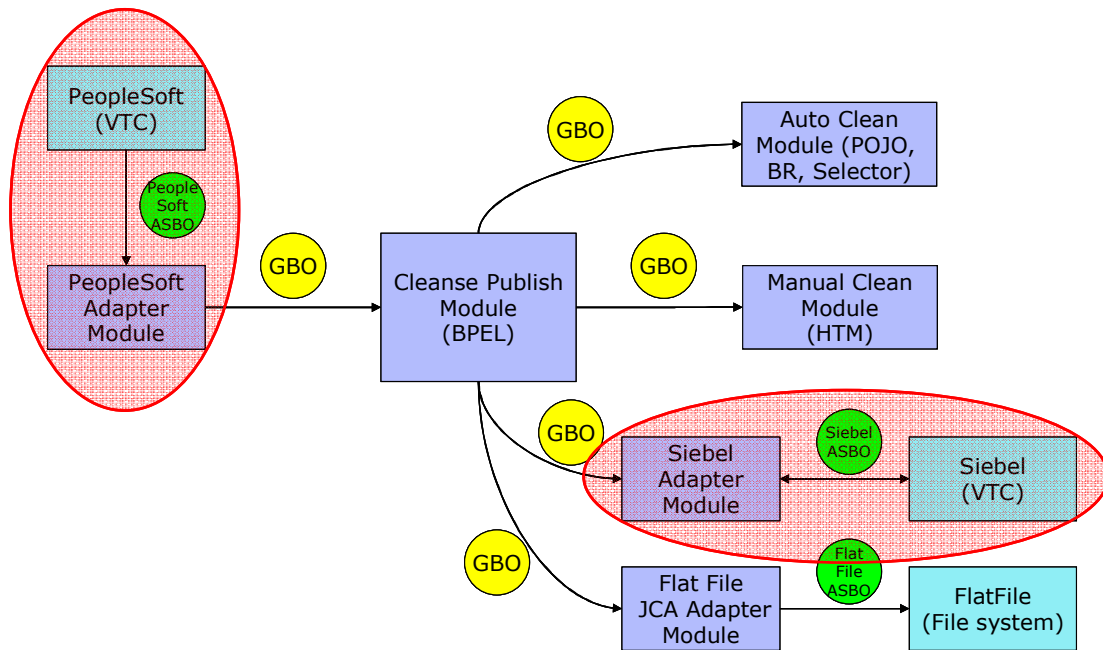
### What you should be able to do

At the end of this lab you should be able to

- Create and Interface map and define the handling of the parameters for the interfaces
- Create a data map with a variety of transforms
- Test the maps using Component Test

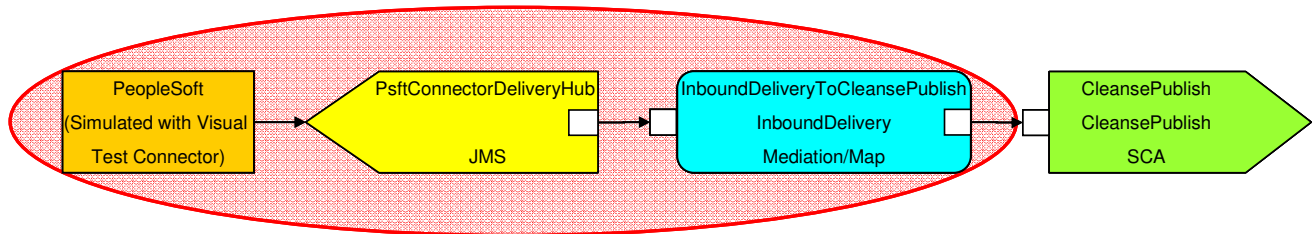
## Introduction

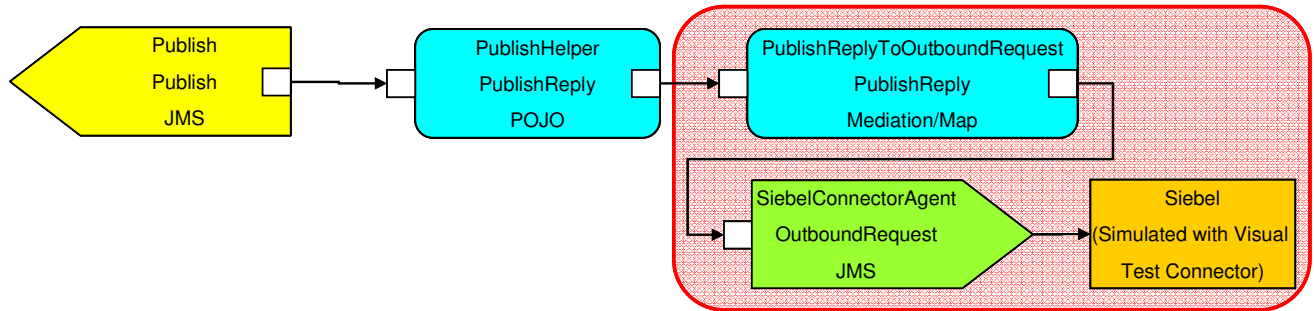
The following diagram highlights the parts of the overall scenario that will be addressed in this lab. You will add mapping of business objects to the PeopleSoft and Siebel modules. This will build upon what you have already developed for these modules in the **WBI Adapters – Import/Export** lab.



## Description of the Module

Data mapping operations are used to convert between ASBOs and GBOs. In WebSphere Integration Developer, Data Maps are always contained within Interface Maps. What you will do is add an Interface Map to each module, and then build the Data Maps that are required to do the PeopleSoft ASBO to GBO mappings and the GBO to Siebel ASBO mappings. Then the test component will be used to test the Interface Map and its contained Data Maps.





## 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\workspaceMaps	/tmp/Labfiles60/eXchange/WBIAdapters/workspace
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles60\WBIAdapters\Solution	/tmp/WBIAdapters/Solution

Windows users' note: 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 **WBIAdapters – Import/Export** 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 avoids any possible problems that may 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 to initialize the Workspace for the lab using the following values:

**<WORKSPACE>**

C:\Labfiles60\exchange\WBIAdapters\workspaceMaps

**<PROJECT\_INTERCHANGE>**

C:\Labfiles60\exchange\WBIAdapters\Solution\ImportExport\_P1.zip

**<MODULE>**

n/a

**<DEPENDENT\_LIBRARIES>**

n/a

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

**<WORKSPACE>**

C:\Labfiles60\exchange\WBIAdapters\workspace

**<PROJECT\_INTERCHANGE>**

n/a

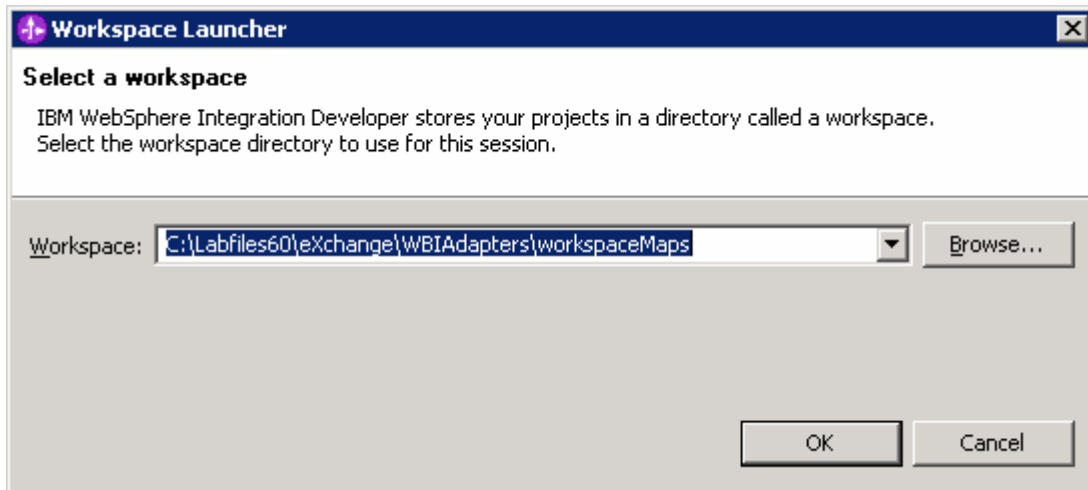
**<MODULE>**

n/a

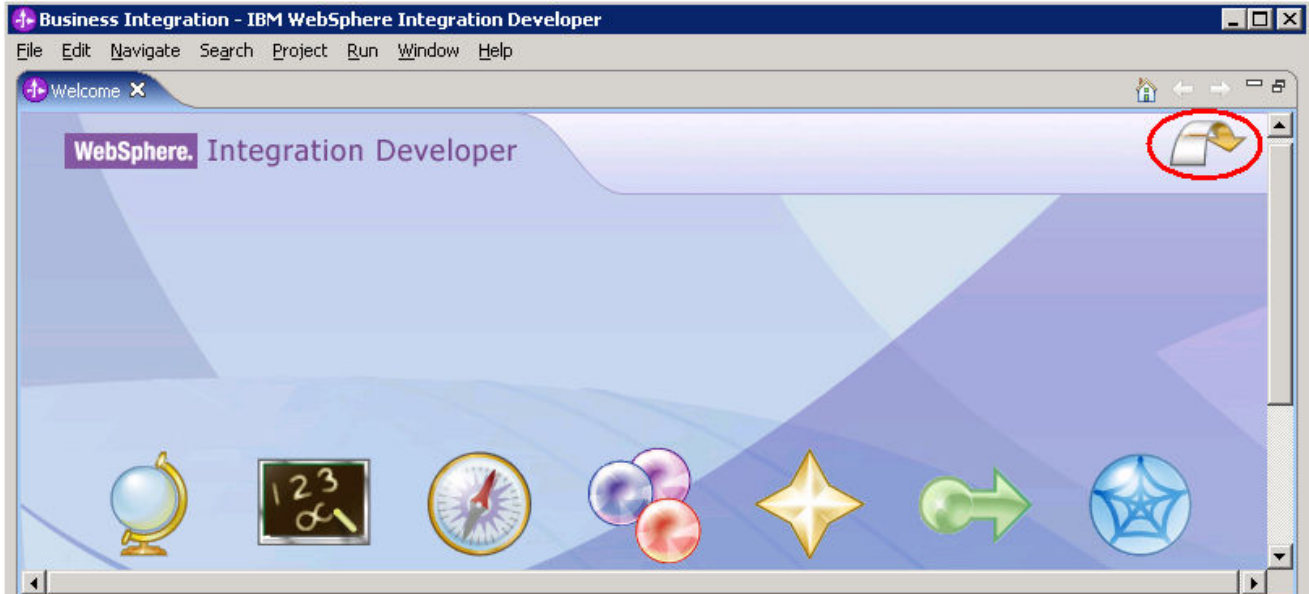
**<DEPENDENT\_LIBRARIES>**

n/a

- \_\_\_ 1. 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).

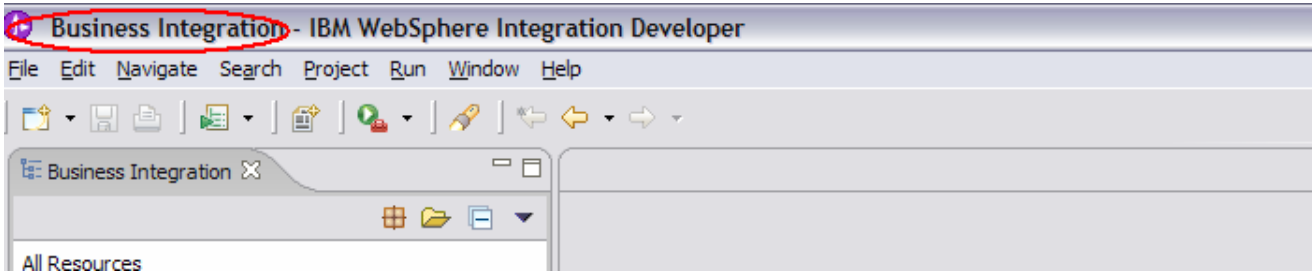


- \_\_\_ d. Ensure you are in the **Business Integration** perspective.

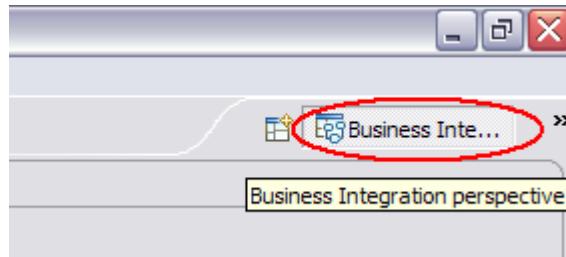
There is more than one way to switch from one perspective to another. The different ways are described below.

- \_\_\_ 1. After WebSphere Integration Developer V6.0 opens, you can determine which perspective you are currently in.

\_\_\_ a. Look for the perspective name in the title bar.



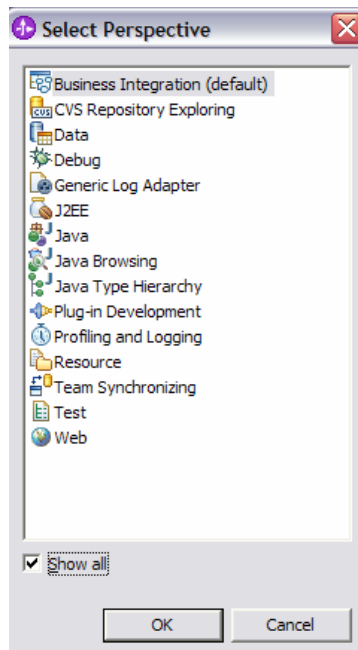
\_\_\_ b. In the top right of WebSphere Integration Developer, you can see the name of the current perspective displayed as you hover the mouse over the perspective icon.



\_\_\_ 2. Select a new perspective you want to switch to.

\_\_\_ c. From the top right of WebSphere Integration Developer workspace, click (🏠) and select Other.

\_\_\_ d. The Select Perspective dialog will appear. Highlight the perspective that you want to open or you can select **Show All** to view all available perspective within the workspace.



\_\_\_ e. Click **OK**.

---

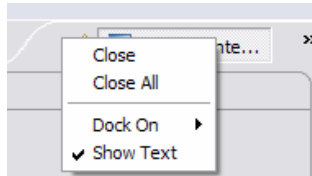
**NOTE:** You can also open perspectives through **Window -> Open Perspective -> Other** and then choose your perspective.

---

- \_\_\_ 3. Select an existing perspective that you have not used before.
- \_\_\_ f. From the top right of WebSphere Integration Developer workspace, click ( >> ) which is next to the name of the current perspective.
- \_\_\_ g. Once you click on ( >> ), you will get the list of existing perspectives that you have opened before from the same workspace.



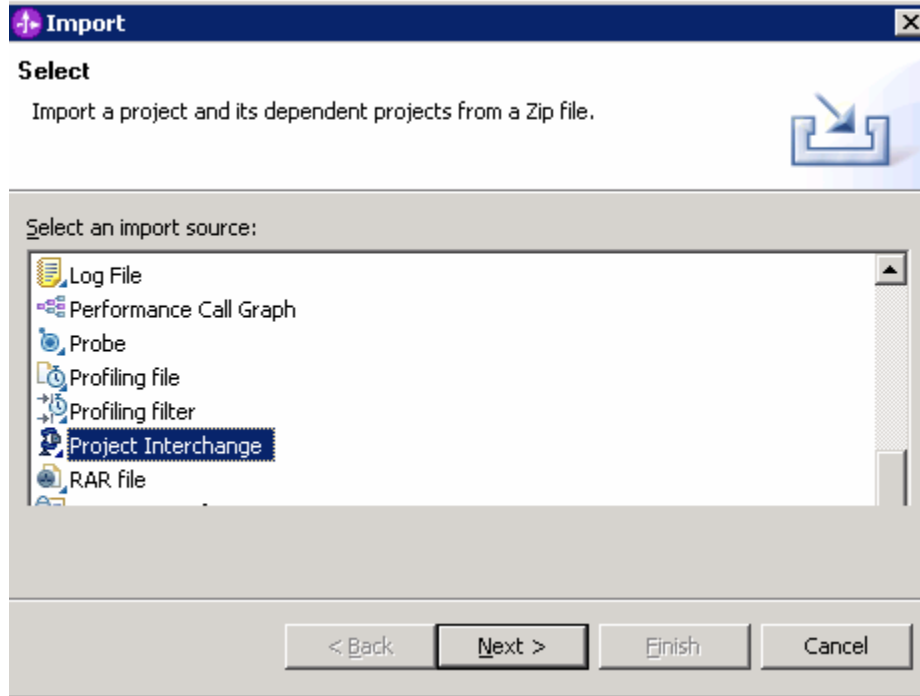
- \_\_\_ 4. Close Perspective.
- \_\_\_ h. Right click on the perspective and select **Close** to close the currently open perspective or **Close All** to close all perspectives.



- \_\_\_ 5. 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>**.
- \_\_\_ i. From the menu bar, select **File -> Import...**



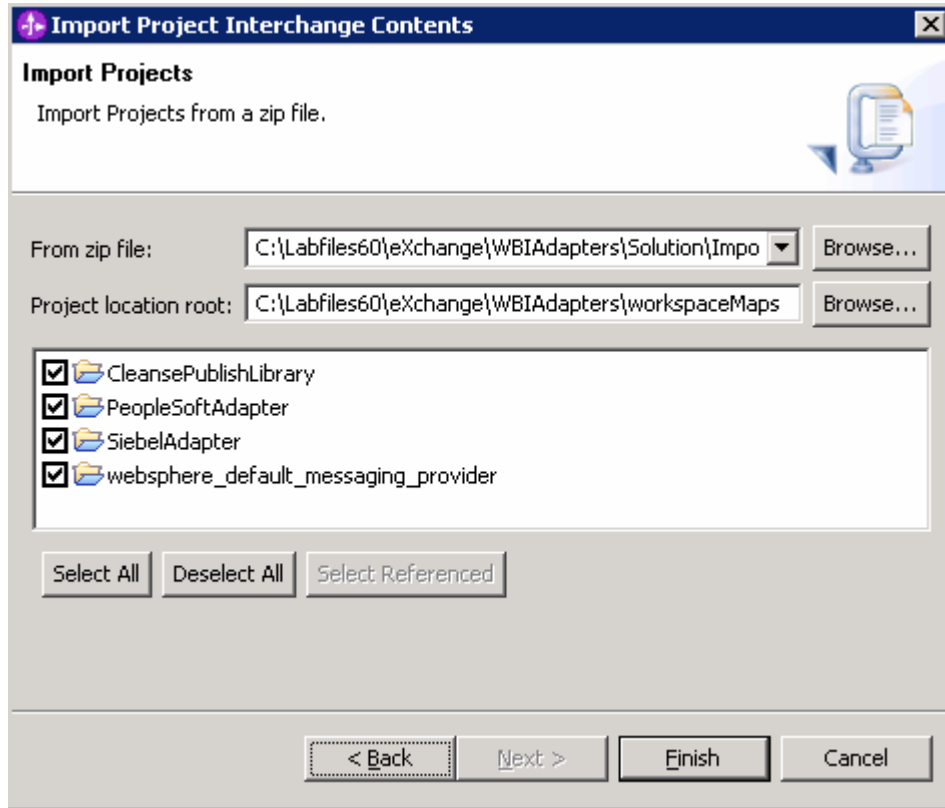
\_\_ j. In the Import dialog, scroll down and select **Project Interchange**.



\_\_ k. Click **Next**.

\_\_ l. In the Import Projects dialog, initialize the From zip file: field to **<PROJECT\_INTERCHANGE>**.

\_\_ m. Click the **Select All** button.

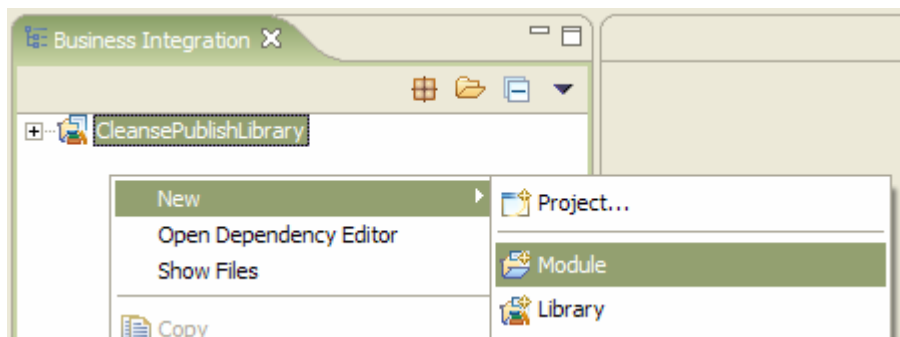


\_\_ n. Click **Finish**.

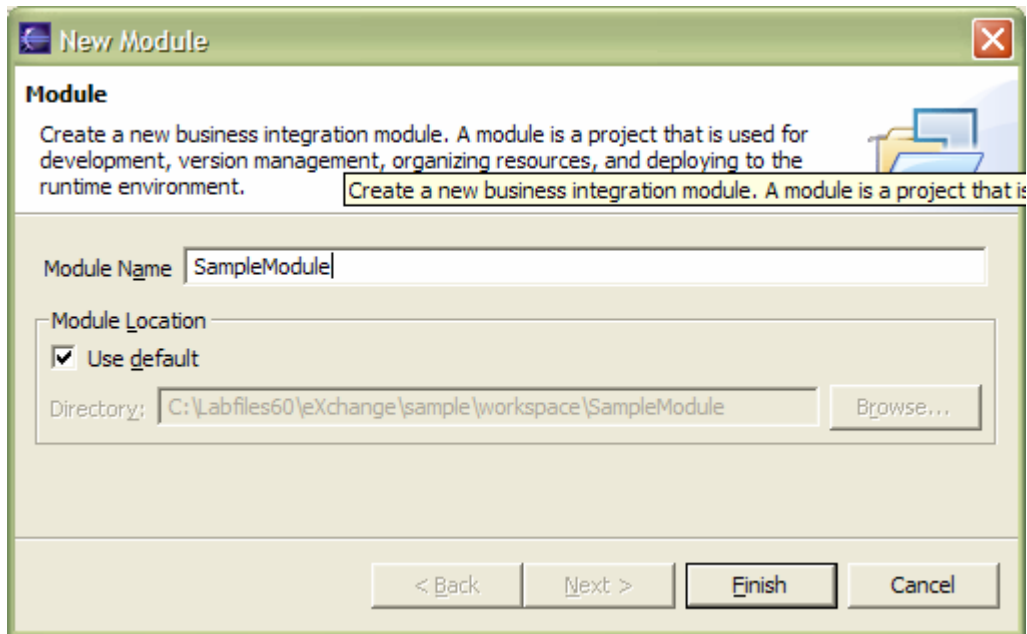
\_\_\_ 6. If this lab requires that you create a Business Integration module called **<MODULE>**, complete these steps:

\_\_ o. Right click on the background of the Business Integration view to access the pop-up menu.

\_\_ p. Select **New > Module**.



\_\_\_ q. In the New Module dialog, enter **<MODULE>** for the Module Name.

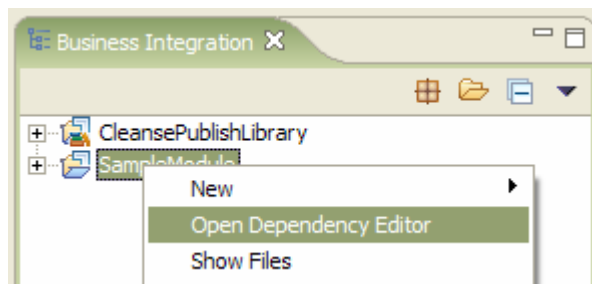


\_\_\_ r. Click **Finish**.

\_\_\_ 7. If this lab requires that **<MODULE>** needs any **<DEPENDENT\_LIBRARIES>**, complete these steps.

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

\_\_\_ t. Select **Open Dependency Editor**.



\_\_\_ u. In the Dependency Editor, click the **Add...** button.

\_\_\_ v. In the Library Selection dialog, select from the list the **<DEPENDENT\_LIBRARIES>**.

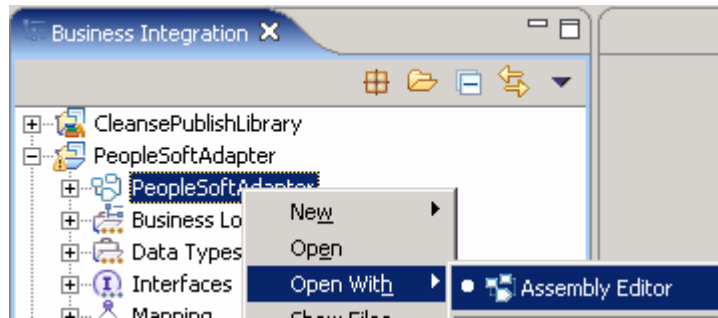
\_\_\_ w. Click **OK**.


\_\_\_ x. Press **Ctrl+S** to save the dependencies for this module.

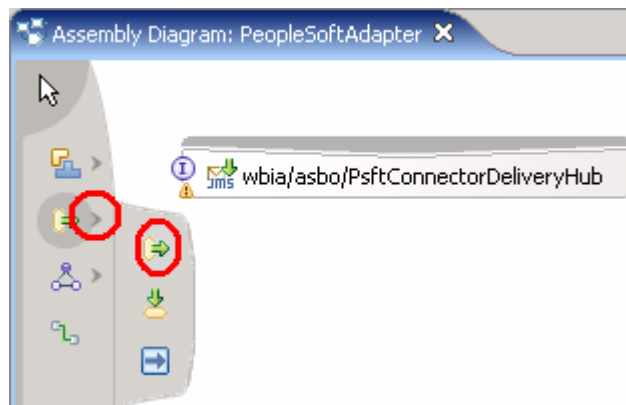
## Part 2: Define Interface Maps and Data Maps for PeopleSoft

In this part you will create the Interface Map needed to map the InboundDelivery interface of the Export for the PeopleSoft adapter to the CleansePublish interface which will be called on the CleansePublish module. Within the Interface Map you will define the Data Maps required to map the PeopleSoft ASBOs to the GBOs.

- \_\_\_ 1. Establish your WebSphere Integration Developer and WebSphere Process Server environments for completing this lab.
  - \_\_\_ a. Start WebSphere Integration Developer specifying a workspace that has been initialized according to the instructions in the **LAB: WBI Adapters – Import/Export**.
  - \_\_\_ b. Ensure you are in the Business Integration perspective (see task **Switching Perspectives within the Workspace**).
- \_\_\_ 8. Open the PeopleSoftAdapter in the Assembly Editor.
- \_\_\_ c. In the Business Integration view, use the + in front of **PeopleSoftAdapter** to expand its sub-tree.
- \_\_\_ d. Right click on the **PeopleSoftAdapter** module assembly and select **Open With > Assembly Editor**.



- \_\_\_ 9. Add the Import that will be used to call the CleansePublish module.
- \_\_\_ e. Click on the (  ) by the second icon down to expand the submenu, and then click on the **Import** icon (icon with the arrow going out).



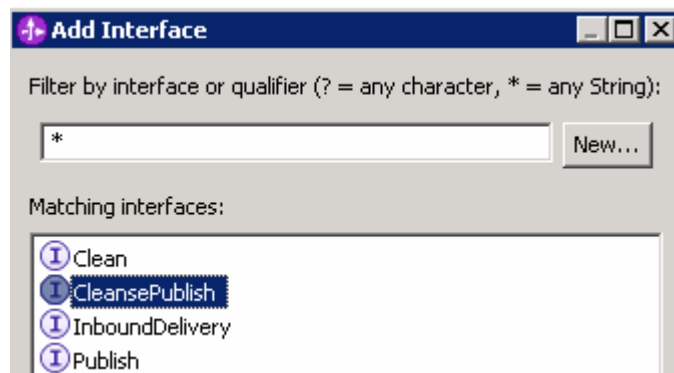
- \_\_\_ f. Move the mouse over the assembly editor pallet and click the mouse to drop the Import, and the result should look like this:



\_\_\_ g. Click on **Import1** and the add Interface icon should appear.

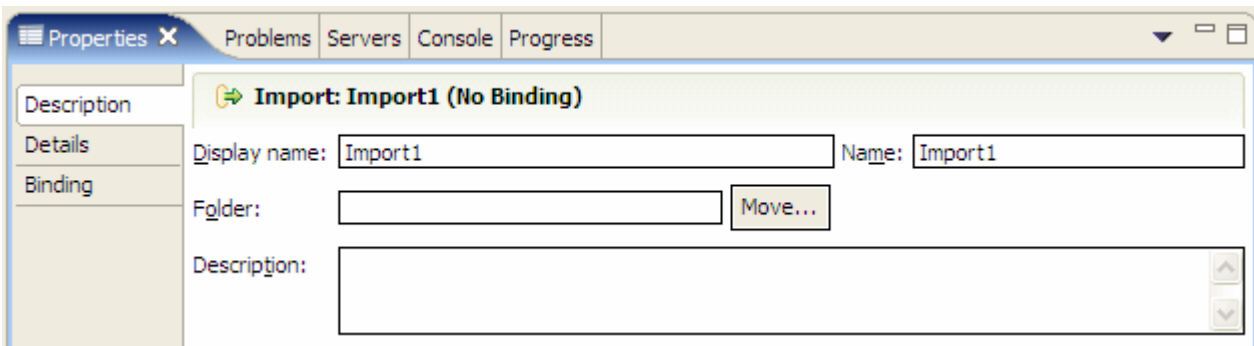


\_\_\_ h. Click on the **Add Interface** icon and the Add Interface dialog will appear.



\_\_\_ i. Select **CleansePublish** and click **OK**.

\_\_\_ j. In the lower panel, click on the **Properties** tab to see the properties panel for Import1.

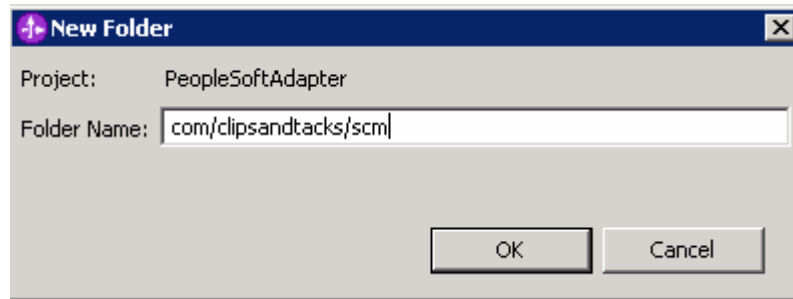


\_\_\_ k. Change the **Display name** and **Name** both to be **CleansePublish** (hint → update the Name field first and the Display name will be automatically updated).

\_\_\_ l. Set the folder by using the **Move...** button.

\_\_\_ m. In the Select Folder for Import CleansePublish dialog, click the **New Folder...** button.

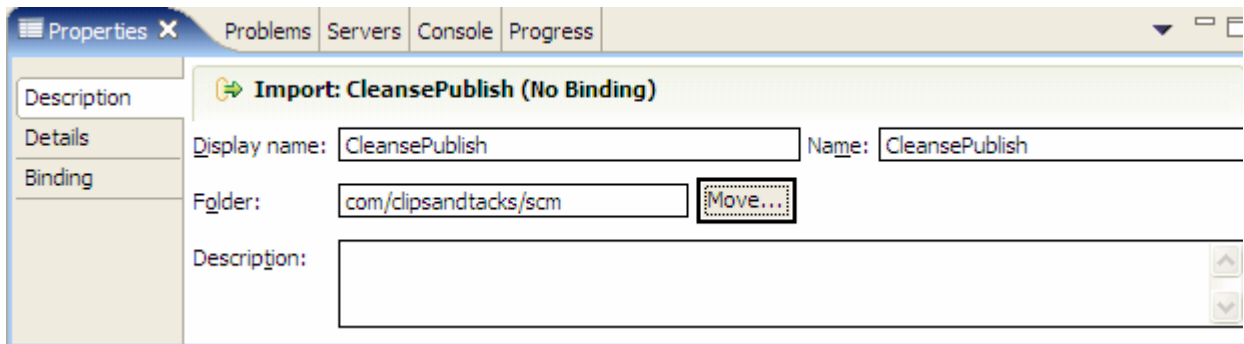
\_\_ n. In the New Folder dialog, enter the folder name **com/clipsandtacks/scm**.



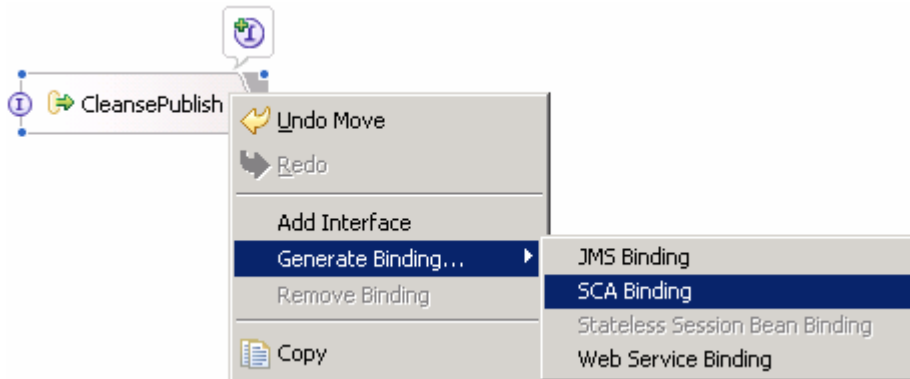
\_\_ o. Click **OK**.

\_\_ p. On the Select Folder for Import CleansePublish dialog, click **OK**.

\_\_ q. Verify that the properties have been correctly initialized to the following values.



\_\_ r. Right click on the **CleansePublish** Import and select **Generate Binding... > SCA Binding**.

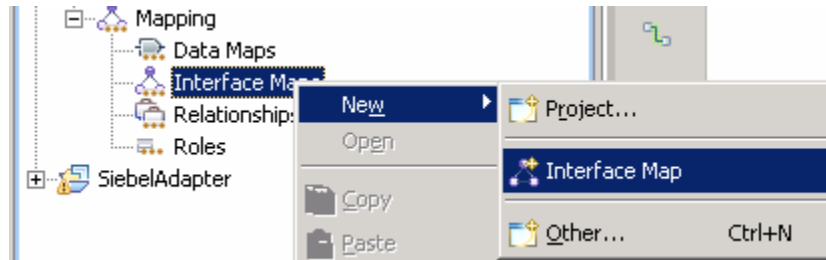


\_\_ s. Press **Ctrl+S** to save the Assembly Diagram.

\_\_\_\_ 10. Add the Interface Map that will map the InboundDelivery interface to the CleansePublish interface.

\_\_ t. In the Business Integration view, expand to **Mapping** under the **PeopleSoftAdapter** sub-tree.

\_\_ u. Right click on **Interface Maps** and select **New > Interface Map**.



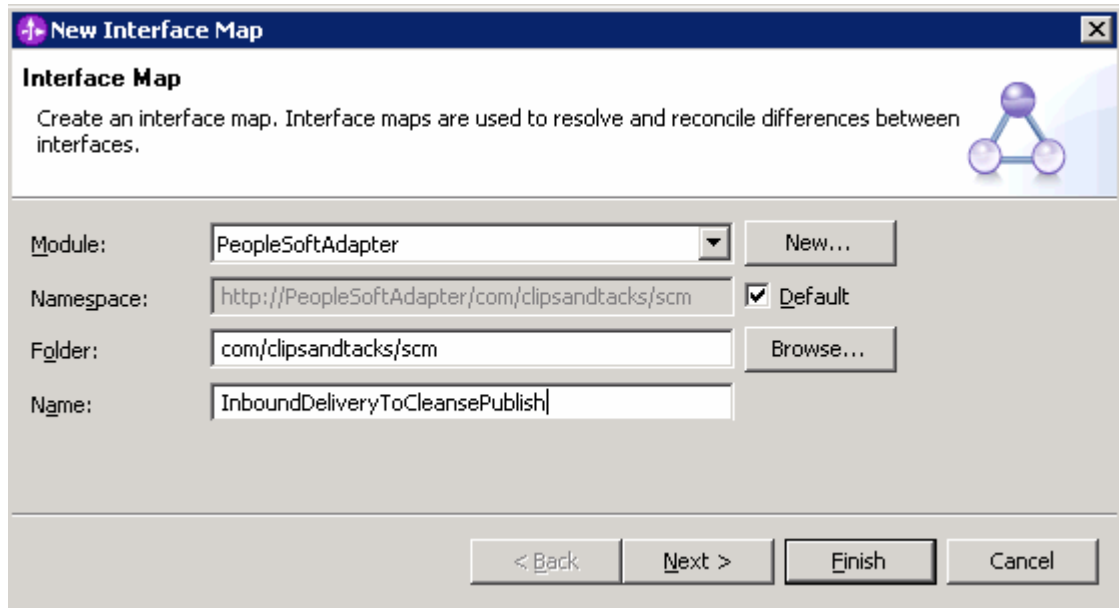
\_\_ v. In the New Interface Map dialog, set the following values:

Module: PeopleSoftAdapter

Namespace: (accept the default)

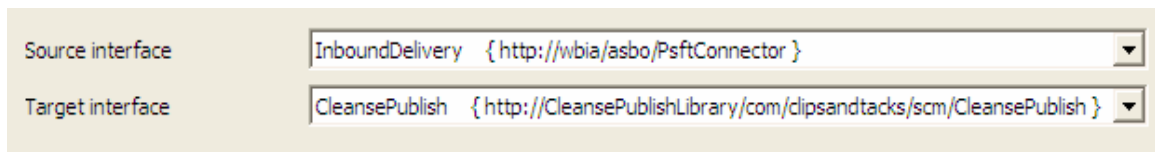
Folder: com/clipsandtacks/scm (you can use the **Browse...** dialog to select this)

Name: InboundDeliveryToCleansePublish

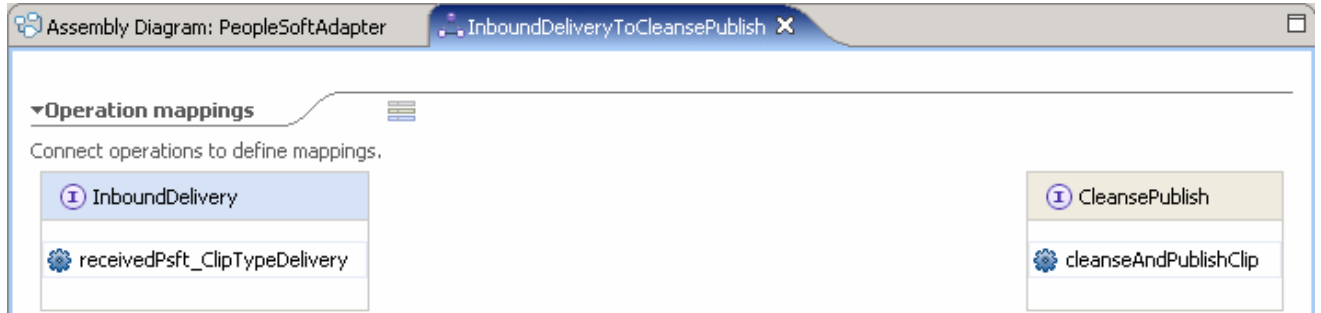


\_\_ w. Click **Next**.

\_\_ x. On the next panel, use the drop down boxes to select **InboundDelivery** as the Source interface and **CleansePublish** as the Target interface.



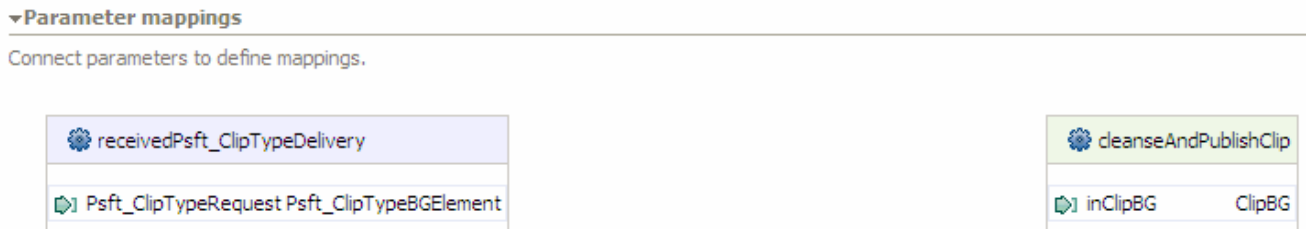
- \_\_\_ y. Click **Finish** and you should see the Interface Mapping editor displayed with the Operation Mappings pane visible.



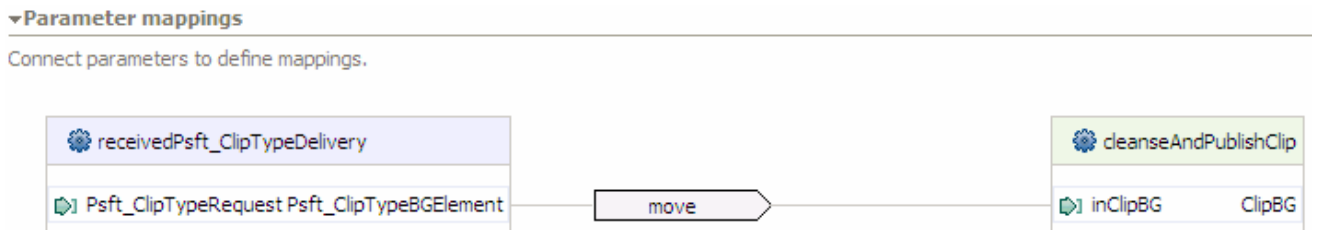
- \_\_\_ 11. Define the InboundDeliveryToCleansePublish Interface Map.
- \_\_\_ z. Draw a line from the receivedPsft\_ClipTypeDelivery operation to the cleanseAndPublishClip operation. This is done by doing a click and hold on **receivedPsft\_ClipTypeDelivery** and dragging and dropping onto **cleanseAndPublishClip**.



- \_\_\_ aa. Click on the line between the two operations, and the Parameter mappings pane will become visible.



- \_\_\_ bb. Draw a line from the **Psft\_ClipTypeRequest** parameter to the **inClipBG** parameter.



- \_\_\_ cc. Notice that the default parameter mapping operation is a **move**. Since these parameters are of different types, you will need to change this to a map in the next step.

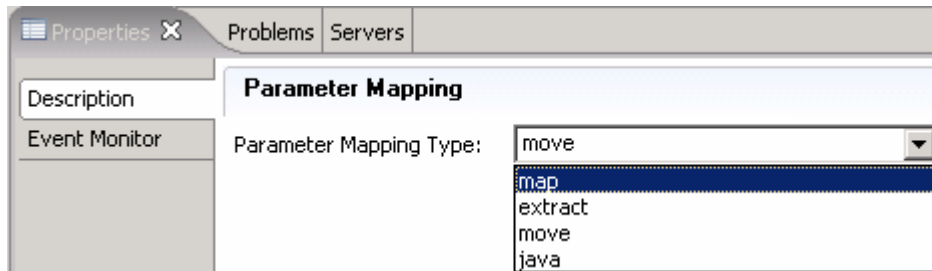


\_\_\_\_ 12. Create the Data Map to map the Psft\_ClipTypeRequest parameter (which is a Psft\_ClipTypeBG) to the inClipBG parameter (which is a ClipBG).

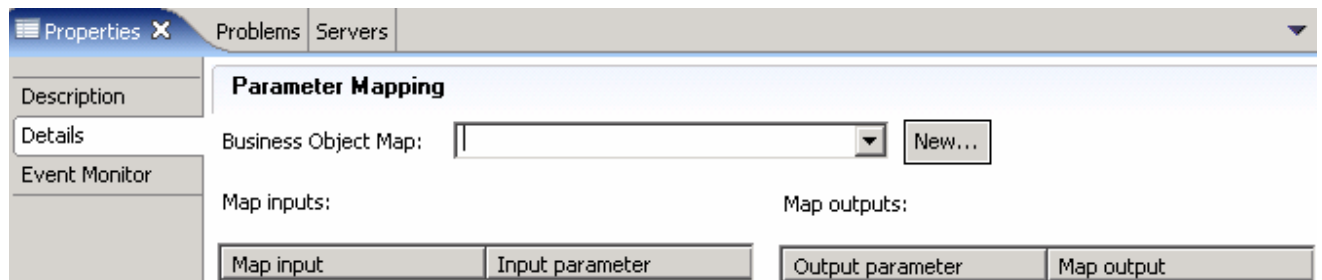
\_\_ dd. Click on the line with the **move** to select it.

\_\_ ee. Click on the **Properties** tab in the lower pane and then on the **Description** tab.

\_\_ ff. Use the drop down box to change the Parameter Mapping Type from move to **map**.



\_\_ gg. Click on the **Details** tab in the Properties view.



\_\_ hh. Click the **New...** button to open the New Business Object Mapping wizard dialog.

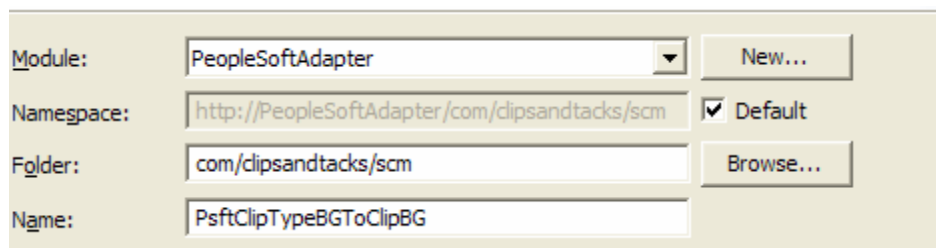
\_\_ ii. In the top portion of the dialog, set the following values.

Module: PeopleSoftAdapter

Namespace: (accept the default)

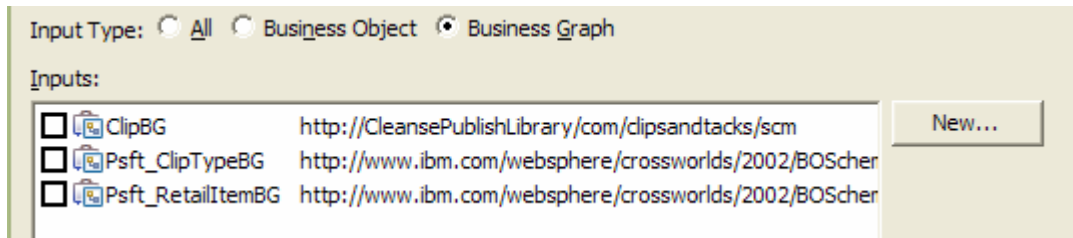
Folder: com/clipsandtacks/scm (you can use the **Browse...** dialog to select this)

Name: PsftClipTypeBGToClipBG



\_\_ jj. Click **Next**.

\_\_ kk. For both the **Input Type** and **Output Type**, click on the **Business Graph** radio button so that the lists are filtered to only show Business Graphs – this makes looking through the list easier.



\_\_ ll. Select **Psft\_ClipTypeBG** for the Inputs.

\_\_ mm. Select **ClipBG** for the Outputs.

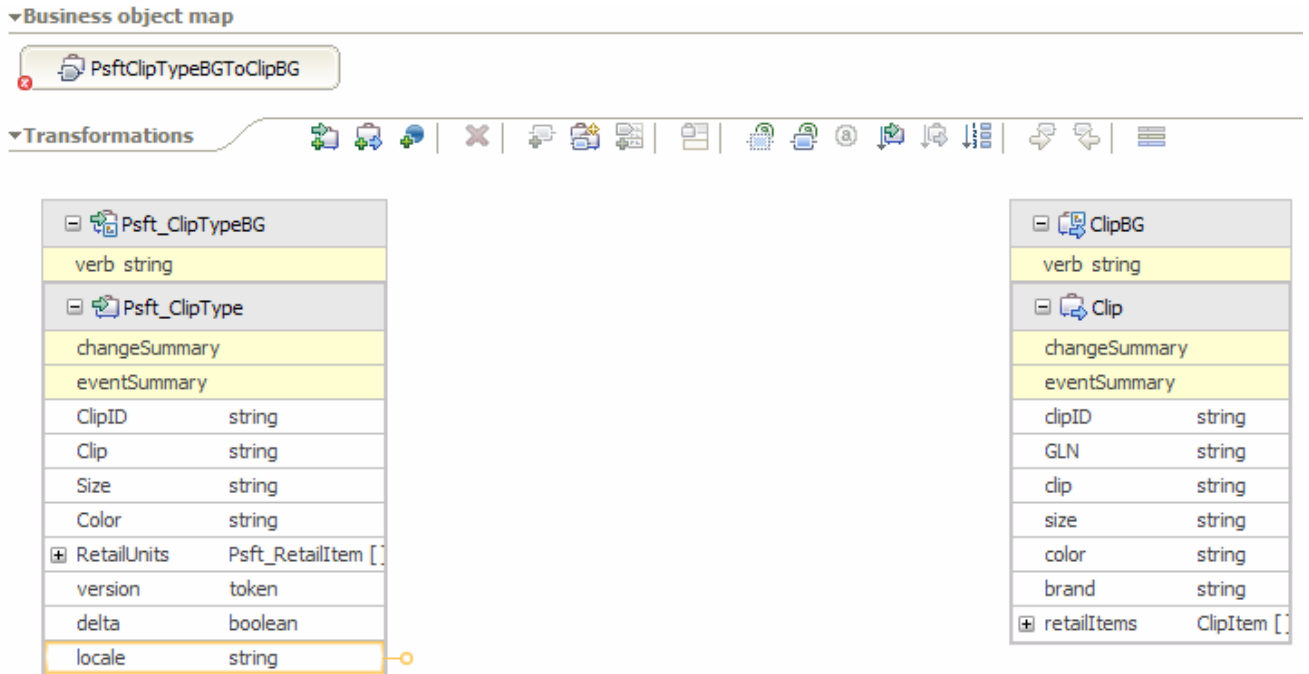
\_\_ nn. Click **Finish**.

\_\_ oo. Press **Ctrl-S** to save the Interface Map.

\_\_\_\_\_ 13. Open the **PsftClipTypeBGToClipBG** map that you just created in the Map Editor.

\_\_ pp. Expand **PeopleSoftAdapter > Mapping > Data Maps** in the Business Integration view.

\_\_ qq. Double click on **PsftClipTypeBGToClipBG** to open it in the Map Editor.



\_\_\_\_\_ 14. Map the verb (when mapping a BG, you generally always want to move the verb).

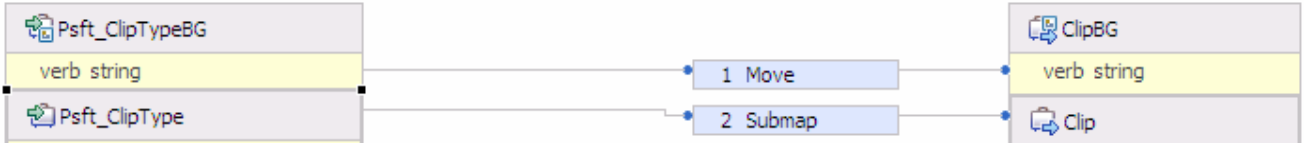
\_\_ rr. Click and hold on the **verb** in Psft\_ClipTypeBG and drag to the **verb** in ClipBG.



\_\_ a. Notice that the default transformation is move, which is what you want in this case.

\_\_\_\_ 15. Map the contained Business Objects.

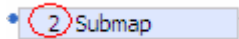
\_\_ b. Click and hold on **Psft\_ClipType** and drag to **Clip**.



\_\_ c. Notice that in this case, since you are mapping from BO to BO, the default transformation is Submap.

\_\_\_\_ 16. Create the Submap.

\_\_ d. Click on the Submap so that you can access information about the submap in the Properties view.



\_\_ e. Click on the **Details** tab in the Properties view.

\_\_ f. Click the **New...** button to open the New Business Object Map wizard dialog.

\_\_ g. In the top portion of the dialog, set the following values.

Module: PeopleSoftAdapter

Namespace: (accept the default)

Folder: com/clipsandtacks/scm

Name: PsftClipTypeToClip

\_\_ h. Click **Next**.

\_\_ i. For both the **Input Type** and **Output Type**, click on the **Business Object** radio button so that the lists are filtered to only show Business Objects – this just makes looking through the list easier.

\_\_ j. For the Inputs, make sure that **Psft\_ClipType** is checked.

\_\_ k. For the Outputs, make sure that **Clip** is checked.

\_\_ l. Click **Finish**. This will put you into the map editor for the map you just created.

▼Business object map

PsftClipTypeToClip

▼Transformations

Psft_ClipType	
ClipID	string
Clip	string
Size	string
Color	string
RetailUnits	Psft_RetailItem [ ]
version	token
delta	boolean
locale	string

Clip	
clipID	string
GLN	string
clip	string
size	string
color	string
brand	string
retailItems	ClipItem [ ]

\_\_\_ 17. Map the fields from Psft\_ClipType to Clip.

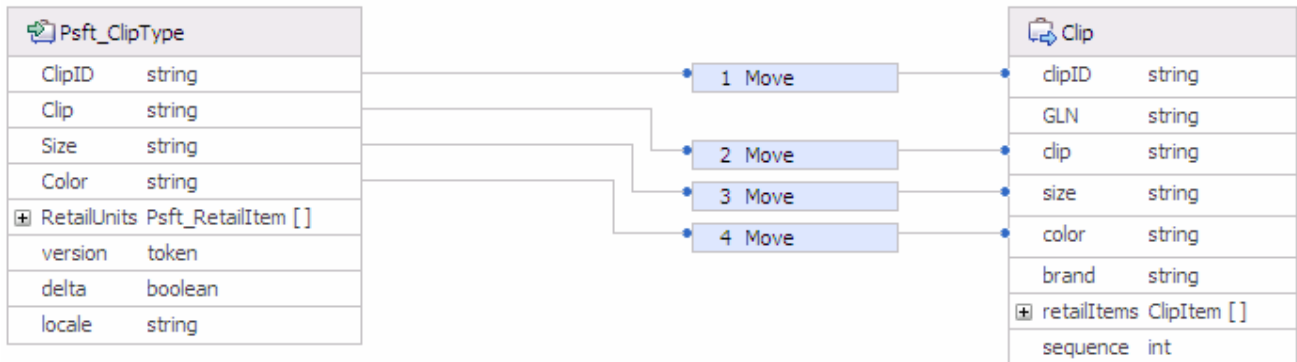
\_\_\_ m. By using the click, hold and drag technique, define move transformations for the following:

ClipID > clipID

Clip > clip

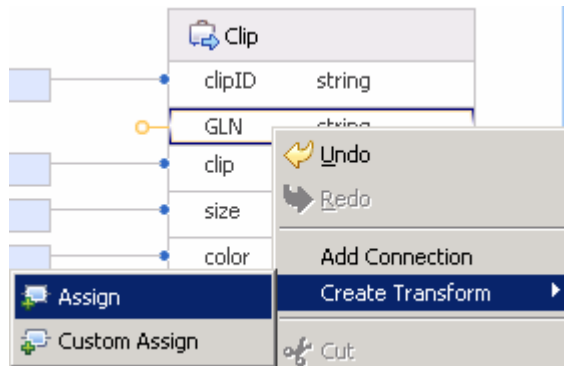
Size > size

Color > color



\_\_\_ n. The GLN field is used to hold the Global Location Number, a 13 digit number which uniquely identifies Clips and Tacks Clip Division. This will be set by the map using an Assign.

1) Right click on **GLN** and select **Create Transform > Assign**.



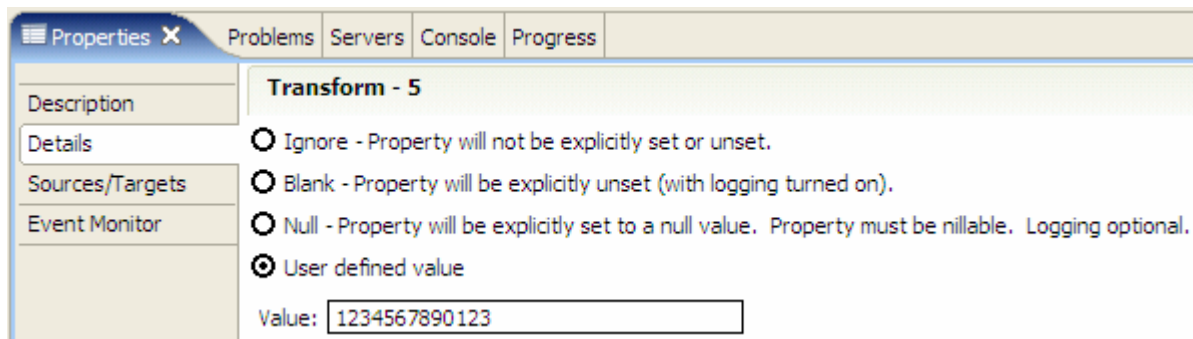
2) Click on the left of the **Assign**.



3) Click on the **Details** tab in the Properties view.

4) Ensure that the **User defined value** radio button is selected.

5) Fill in a value of **1234567890123**.



\_\_\_ o. The brand field is also set by the map to Clips and Tacks.

1) Right click on brand and select **Create Transform > Assign**.

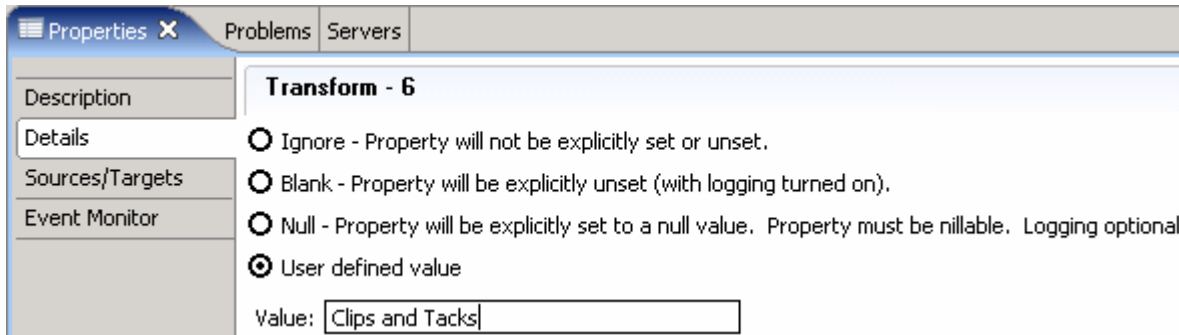
2) Click on the left of the **Assign**.



3) Click on the **Details** tab in the Properties view.

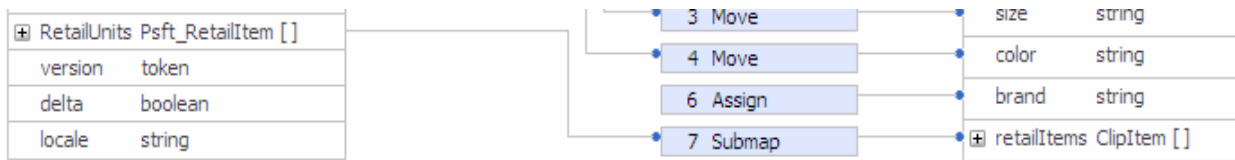
4) Ensure that the **User defined value** radio button is selected.

5) Fill in a value of **Clips and Tacks**.



\_\_ p. Map RetailUnits to retailItems. This will require a submap.

1) Drag a line between RetailUnits Psft\_RetailItem [ ] and retailItems ClipItem [ ] to define a submap transform.



2) Click on the left of the **Submap**.

3) In the Properties view, click on the **Details** tab.

4) Click the **New...** button by the Business Object Mapping field.

5) In the top portion of the dialog, set the following values.

Module: PeopleSoftAdapter

Namespace: (accept the default)

Folder: com/clipsandtacks/scm

Name: PsftRetailItemToClipItem

6) Click **Next**.

7) For both the **Input Type** and **Output Type**, click on the **Business Object** radio button so that the lists are filtered to only show Business Objects.

8) For the Inputs, ensure that **Psft\_RetailItem** is selected.

9) For the Outputs, ensure that **ClipItem** is selected.

10) Click **Finish**. This will put you into the map editor for the map you just created:

▼Business object map

PsftRetailItemToClipItem

▼Transformations



Psft_RetailItem	
ItemCode	string
Package	string
Quantity	string
Price	string
AvailableDate	string
WithdrawDate	string
OwningContact	string
version	token
delta	boolean
locale	string

ClipItem	
itemID	string
GTIN	string
package	string
quantity	string
fullDescription	string
price	string
startDate	string
endDate	string
contactFirstName	string
contactLastName	string

\_\_\_ 18. Map the fields from Psft\_RetailItem to ClipItem.

\_\_\_ a. By using the click, hold and drag technique, define move transformations for the following:

ItemCode > itemID

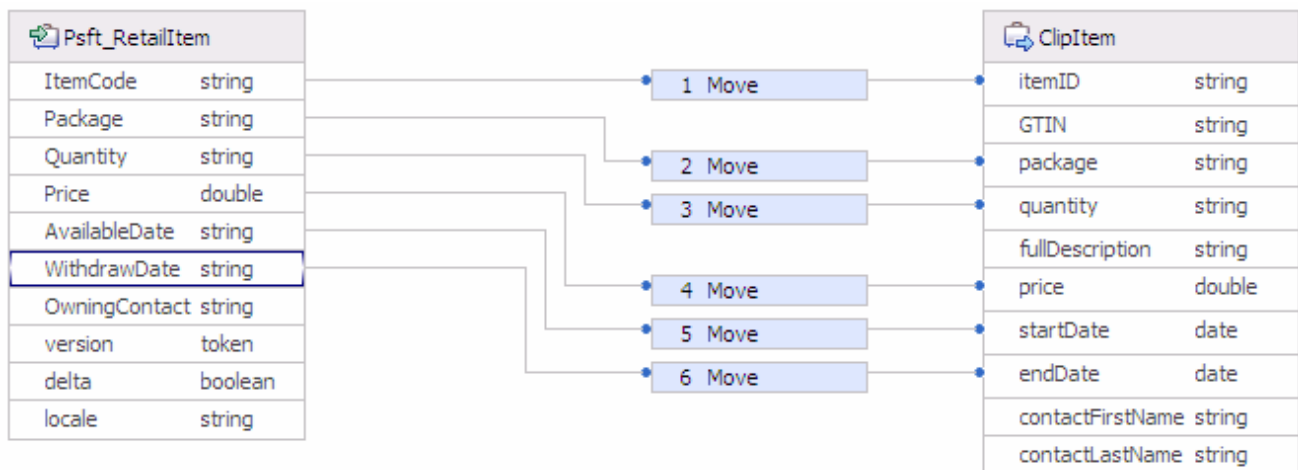
Package > package

Quantity > quantity

Price > price

AvailableDate > startDate

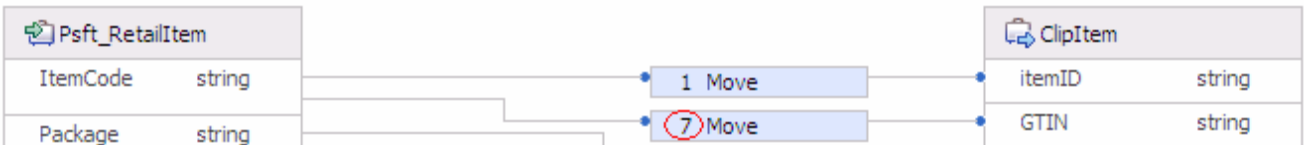
WithdrawDate > endDate



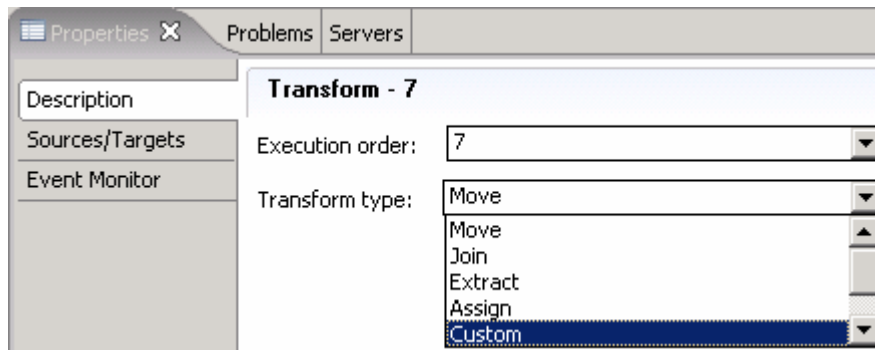
\_\_\_ b. GTIN is a Global Trade Item Number is a 14 digit number which uniquely identifies this saleable item. It is composed of a portion that is unique to ClipsAndTacks.com and a portion that is a unique identifier within ClipsAndTacks.com. This will be accomplished using a custom transform

containing a Java snippet that concatenates a constant for the ClipsAndTacks.com portion and the item code for this retail item.

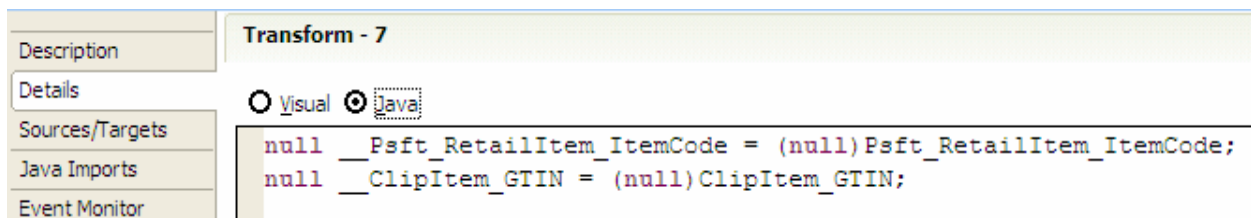
- 1) Draw a line from **ItemCode** to **GTIN** which will create a move transform.
- 2) Click on the left side of the Move.



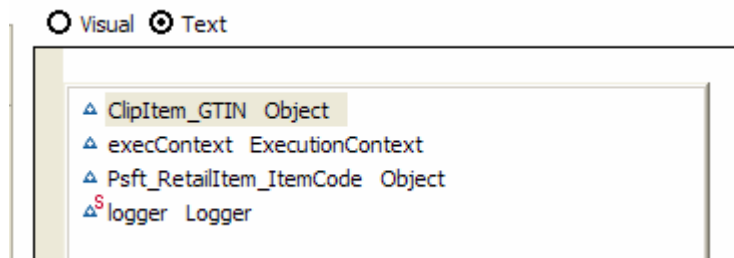
- 3) In the Properties view, on the **Description** tab, use the Transform type dropdown box to select **Custom**.



- 4) Click on the **Details** tab.
- 5) Select the **Java** radio button.
- 6) Click on **Yes** for the Question dialog that pops up.
- 7) The snippet code that is displayed is not needed. Select it (Ctrl+A) and click **Delete** to remove it.

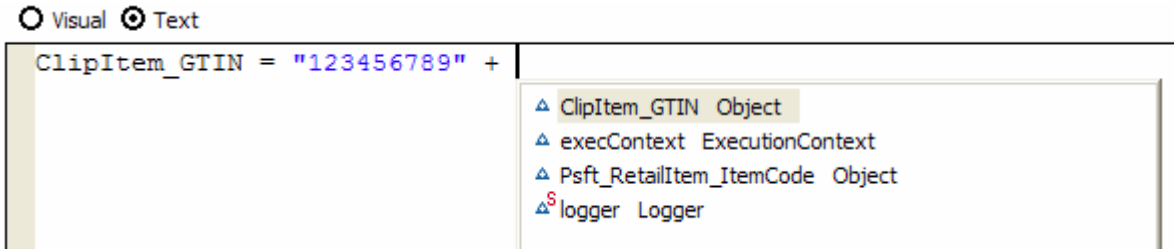


- 8) With the cursor placed in the text entry box, click **Ctrl-Spacebar** to get the context assist dropdown.

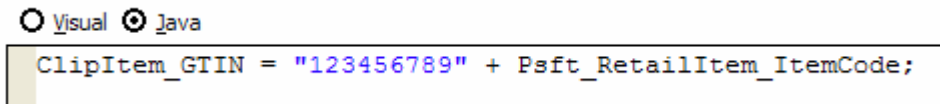




- 9) Select the target field **ClipItem\_GTIN** and click **Enter**.
- 10) Enter a **space**, **=**, a **space**, **"123456789"**, a **space**, **+**, a **space** and then click **Ctrl-Spacebar** again for context assist.

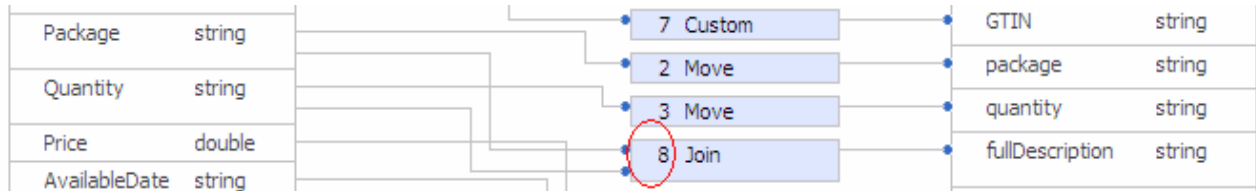


- 11) Select the field **Psft\_Retailitem\_ItemCode** and click **Enter**.
- 12) Enter **;** (**semi colon**) and result should look like this:

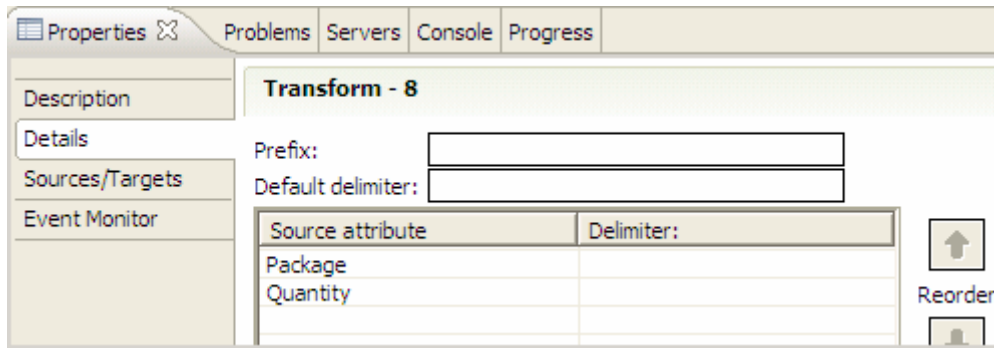


\_\_ c. Handle the fullDescription field by combining the Package and Quantity fields using a Join transform.

- 1) Drag a line from **Package** to **fullDescription** and you will see a Move transform generated.
- 2) Drag a line from **Quantity** to **fullDescription** and you will see the transform changed to a Join.

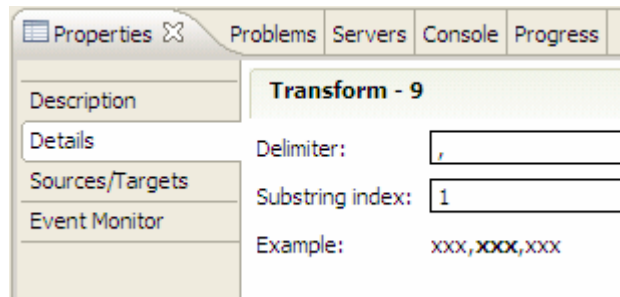


- 3) Click on the left of the **Join** to bring up this transform in the Properties view.
- 4) Click on the **Details** tab in the **Properties** view. You will see the fields that are part of the Join. You will also notice that there is the ability to reorder the fields and to specify a delimiter to be inserted between the fields. In this case you will use a blank for the delimiter and the order is correct, so there is nothing else to do.

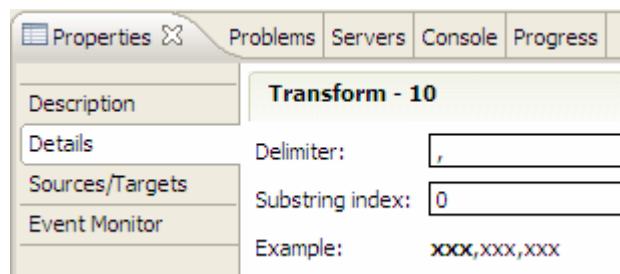


\_\_\_ d. Handle the **OwningContact** field, which is in the format of lastname,firstname. This field must be split between the **contactFirstName** and **contactLastName** fields which can be done using the **Extract** transform.

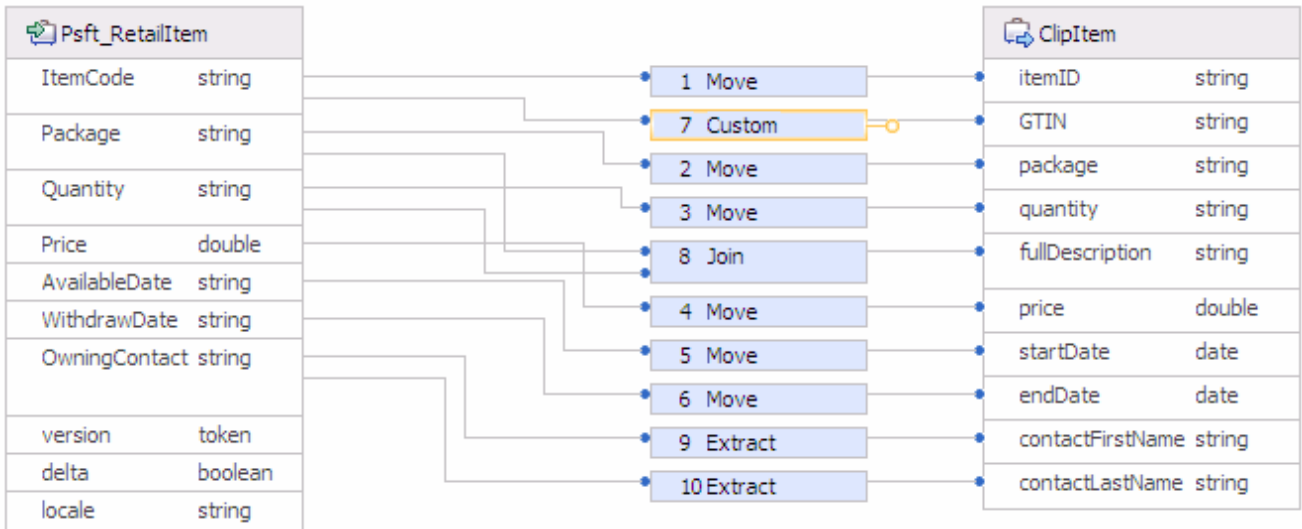
- 1) Draw a line from **OwningContact** to **contactFirstName** and the default Move transform will be created.
- 2) Click in front of the Move to bring up the transform in the Properties view.
- 3) Click on the **Description** tab.
- 4) In the **Transform Type** field, use the dropdown box and select **Extract**.
- 5) Click on the **Details** tab.
- 6) Enter a , (comma) for the delimiter.
- 7) Enter a 1 for the **Substring index** (this is a zero based number, so 1 is the second item). You will notice at the bottom there is an example which shows the delimiter and the substring index (position) you have defined.



- 8) Draw a line from **OwningContact** to **contactLastName** and the default Move transform will be created.
- 9) Click in front of the Move to bring up the transform in the Properties view.
- 10) Click on the **Description** tab.
- 11) In the **Transform Type** field, use the dropdown box and select **Extract**.
- 12) Click on the **Details** tab.
- 13) Enter a , (comma) for the delimiter.
- 14) Enter a 0 for the position.



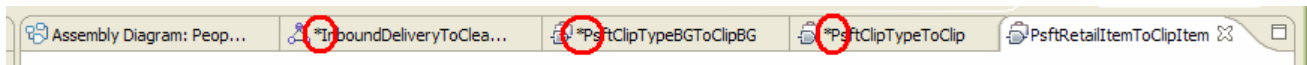
\_\_\_ e. Your map should look like this:



\_\_\_ f. Press **Ctrl-S** to save the map.

\_\_\_ 19. The Interface Map and all three Data Maps are now complete. Prior to proceeding, ensure that all the maps you have been working with are saved.

\_\_\_ q. Examine the tabs at the top of the editors pane.



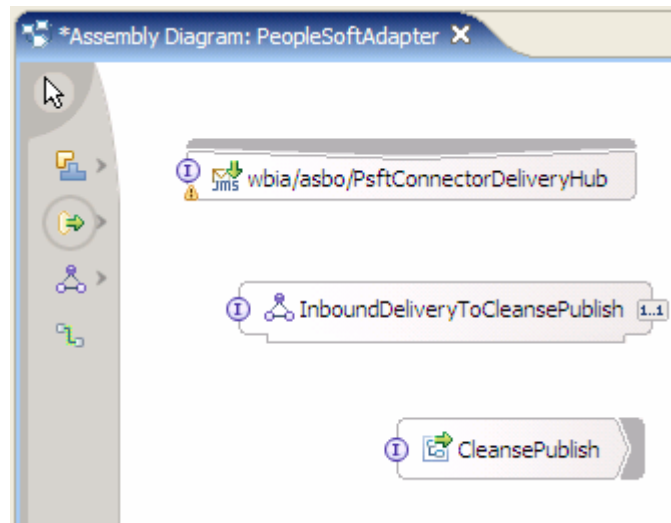
\_\_\_ r. For each one with an asterisk (\*), click on the tab to bring that editor to the foreground and then click **Ctrl-S** to save.

\_\_\_ 20. You are now ready to add the Interface Map to the wiring diagram.

\_\_\_ a. Click on the **Assembly Diagram: PeopleSoftAdapter** tab on the top of the editors pane.

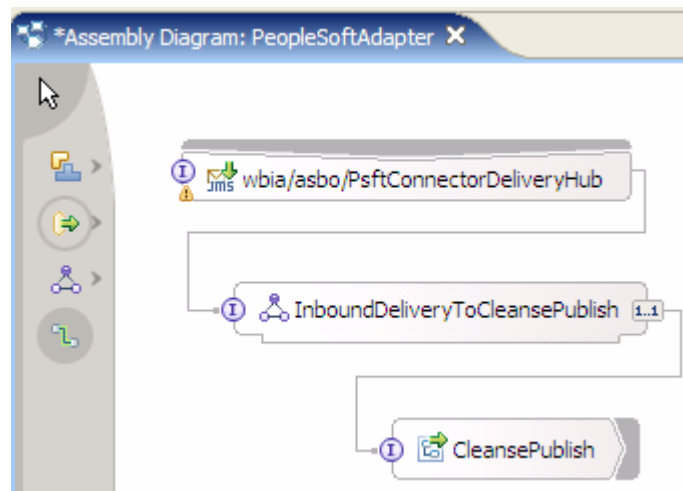
\_\_\_ b. In the Business Integration view, navigate to **PeopleSoftAdapter > Mapping > Interface Maps**.

\_\_\_ c. Click on **InboundDeliveryToCleansePublish** and drag it onto the Assembly Editor.



\_\_\_ d. Draw a wire from the **PsfConnectorDeliveryHub** Export to the **InboundDeliveryToCleansePublish** Interface Map.

\_\_\_ e. Draw a wire from the **InboundDeliveryToCleansePublish** Interface Map to the **CleansePublish** Import.



\_\_\_ f. Press **Ctrl-S** to save the Assembly Diagram.

## Part 3: Define Interface maps and Data maps for Siebel

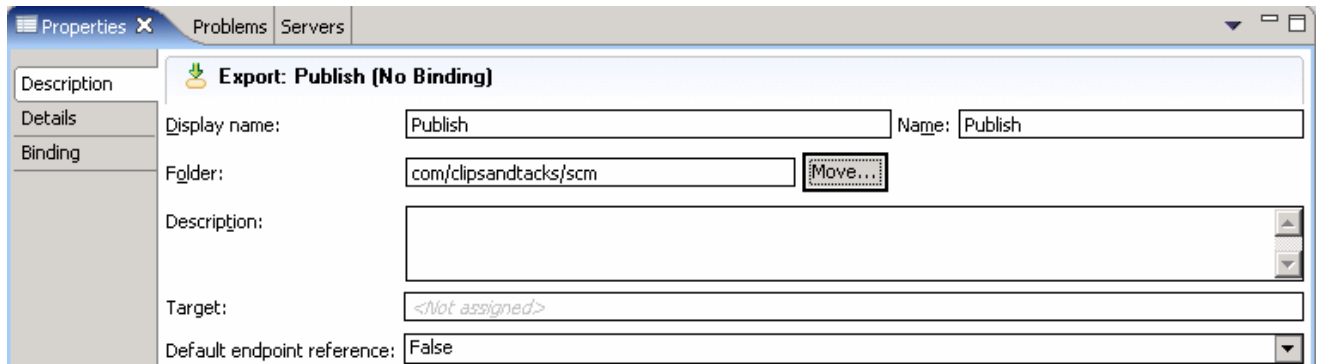
In this part you will create the Interface Map needed to map the Publish interface of the Export which is called by the CleansePublish module to the OutboundRequest interface supported but the Import used to talk to the Siebel Adapter. Within the Interface Map you will define the Data Maps required to map the GBOs to the Siebel ASBOs.

- \_\_\_ 1. Open the SiebelAdapter in the Assembly Editor.
  - \_\_\_ a. In the Business Integration view, Expand **SiebelAdapter**.
  - \_\_\_ b. Right click on the **SiebelAdapter** module assembly and select **Open With > Assembly Editor**.
- \_\_\_ 21. Add the **Export** that will be used to call this module from the CleansePublish module.
  - \_\_\_ a. Click on the > by the second icon from the top to expand the submenu, then click on the **Export** icon (the arrow coming into the icon).



- \_\_\_ b. Move the mouse over the assembly editor pallet and click the mouse to drop the **Export**.
- \_\_\_ c. Click on **Export1** and the add Interface icon should appear.
- \_\_\_ d. Click on the **Add Interface** icon and the Add Interface dialog will appear.
- \_\_\_ e. Select **Publish** and click **OK**.
- \_\_\_ f. Click on the **Properties** tab in the lower panel to see the properties panel for Export1.
- \_\_\_ g. Change the **Display name** and **Name** both to **Publish**.
- \_\_\_ h. Set the location by using the **Move...** button.
- \_\_\_ i. Click the **New Folder...** button in the **Select Folder for Export Publish** dialog.
- \_\_\_ j. Enter the folder name **com/clipsandtacks/scm** in the **New Folder** dialog.
- \_\_\_ k. Click **OK**.
- \_\_\_ l. Click **OK** on the **Select Folder for Export Publish** dialog.

\_\_\_ m. Verify that the properties have been correctly initialized to the following values.



\_\_\_ n. Press **Ctrl-S** to save the Assembly Diagram.

\_\_\_ 22. Add the Interface Map that will map the Publish interface to the OutboundRequest interface.

\_\_\_ a. In the Business Integration view, Expand **Mapping** under the **SiebelAdapter** sub tree.

\_\_\_ b. Right click on Interface Maps and select **New > Interface Map**.

\_\_\_ c. In the **New Interface Map** dialog, set the following values:

Module: SiebelAdapter

Namespace: (accept the default)

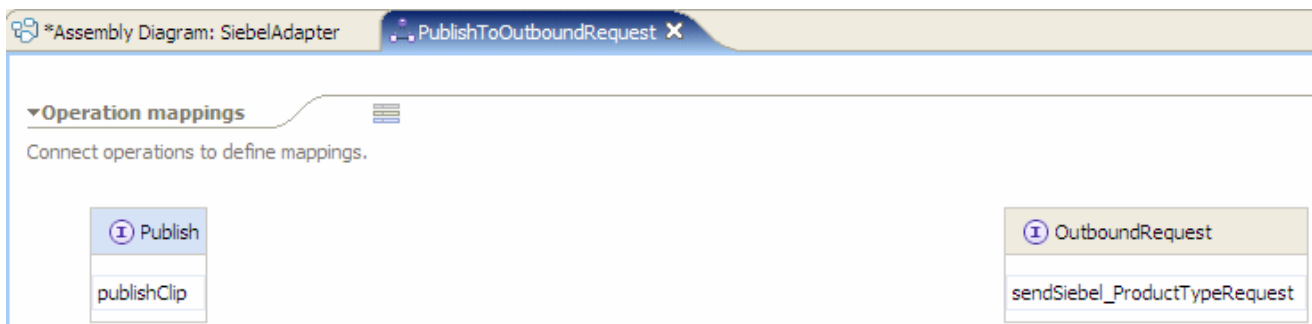
Folder: com/clipsandtacks/scm (you can use the **Browse...** dialog to select this.)

Name: PublishToOutboundRequest

\_\_\_ d. Click **Next**.

\_\_\_ e. On the next panel, use the drop down boxes to select **Publish** as the Source interface and **OutboundRequest** as the Target interface.

\_\_\_ f. Click **Finish** and you should see the Interface Mapping editor displayed with the Operation Mappings pane visible.



\_\_\_ 23. Define the PublishToOutboundRequest Interface Map.

\_\_\_ a. Draw a line from the **publishClip** operation to the **sendSiebel\_ProductTypeRequest** operation.

- \_\_\_ b. Click on the line between the two operations, and the Parameter mappings pane will become visible.

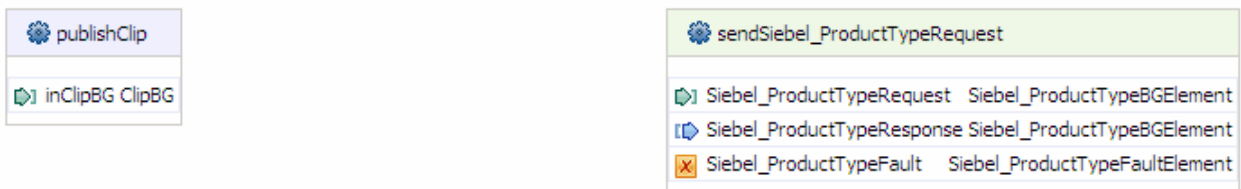
▼ **Operation mappings**

Connect operations to define mappings.



▼ **Parameter mappings**

Connect parameters to define mappings.



NOTE: You will notice that the publishClip operation only has an input while the sendSiebel\_ProductTypeRequest not only has an input but also has an output and a fault. In this lab you will just map the inputs and ignore the output and the fault. In the subsequent lab on relationships changes will be made to handle the output.

- \_\_\_ c. Draw a line from the **inClipBG** parameter to the **Siebel\_ProductTypeRequest** parameter.
- \_\_\_ d. Notice that the default parameter mapping operation is a **move**. Since these parameters are of different types, you will need to change this to a map in the next step.
- \_\_\_ 24. Create the Data Map to map the inClipBG parameter (which is a ClipBG) to the Siebel\_ProductTypeRequest parameter (which is a Siebel\_ProductTypeBG).
- \_\_\_ a. Click on the **Move**.
- \_\_\_ b. Click on the **Properties** tab in the lower pane and then on the **Description** tab.
- \_\_\_ c. Use the drop down box to change the Parameter Mapping Type from move to **map**.
- \_\_\_ d. Click on the **Details** tab in the Properties view.
- \_\_\_ e. Click the **New...** button to open the **New Business Object Map** wizard dialog.
- \_\_\_ f. Set the following values in the top portion of the dialog:
- Module: SiebelAdapter
  - Namespace: (accept the default)
  - Folder: com/clipsandtacks/scm (you can use the **Browse...** dialog to select this)
  - Name: ClipBGTtoSiebelProductTypeBG
- \_\_\_ g. Click **Next**.

- \_\_\_ h. For both the **Input Type** and **Output Type**, click on the **Business Graph** radio button so that the lists are filtered to only show Business Graphs – this just makes looking through the list easier.
- \_\_\_ i. For the Inputs, select **ClipBG**.
- \_\_\_ j. For the Outputs, select **Siebel\_ProductTypeBG**.
- \_\_\_ k. Click **Finish**.
- \_\_\_ l. Press **Ctrl-S** to save the Interface Map.

\_\_\_ 25. Open the **ClipBGToSiebelProductTypeBG** map that you just created in the Map Editor.

\_\_\_ a. Expand **SiebelAdapter > Mapping > Data Maps** in the **Business Integration** view.

\_\_\_ b. Double click on **ClipBGToSiebelProductTypeBG** to open it in the Map Editor.

\_\_\_ 26. Map the verb.

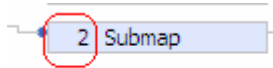
\_\_\_ a. Click and hold on the **verb** in **ClipBG** and drag to the **verb** in **Siebel\_ProductTypeBG** which will create a **Move** transform.

\_\_\_ 27. Map the contained Business Objects.

\_\_\_ a. Click and hold on **Clip** and drag to **Siebel\_ProductType** which will create a **Submap** transform.

\_\_\_ 28. Create the Submap.

\_\_\_ a. Click on the **Submap** so that you can access information about the **Submap** in the **Properties** view.



\_\_\_ b. Click on the **Details** tab in the **Properties** view.

\_\_\_ c. Click the **New...** button to open the **New Business Object Map** wizard dialog.

\_\_\_ d. Set the following values in the top portion of the dialog:

Module: SiebelAdapter

Namespace: (accept the default)

Folder: com/clipsandtacks/scm

Name: ClipToSiebelProductType

\_\_\_ e. Click **Next**.

\_\_\_ f. For both the **Input Type** and **Output Type**, click on the **Business Object** radio button so that the lists are filtered to only show Business Objects.

\_\_\_ g. For the **Inputs**, select **Clip**.

\_\_\_ h. For the **Outputs**, select **Siebel\_ProductType**.

\_\_\_ i. Click **Finish**. This will put you into the map editor for the map you've just created.



▼ Business object map

ClipToSiebelProductType

▼ Transformations



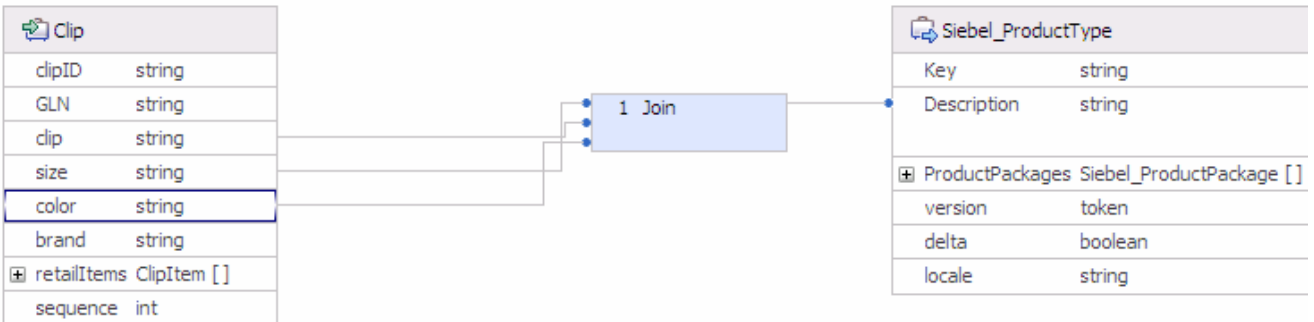
Clip	
clipID	string
GLN	string
clip	string
size	string
color	string
brand	string
retailItems	ClipItem []

Siebel_ProductType	
Key	string
Description	string
ProductPackages	Siebel_ProductPackage []
version	token
delta	boolean
locale	string

\_\_\_ 29. Map the fields from Clip to Siebel\_ProductType.

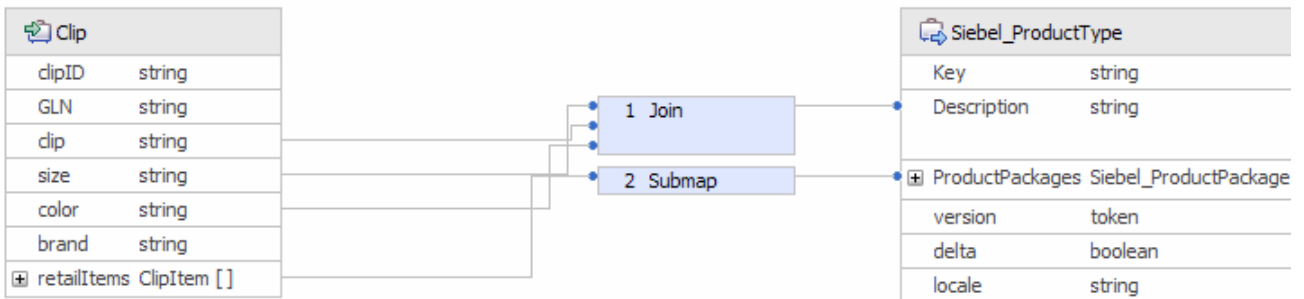
\_\_\_ j. The first transform will be to use a Join to initialize the Description field with the size, clip and color fields.

- 1) Draw a line from **size** to **Description**. This will result in the default Move transformation.
- 2) Draw a line from **clip** to **Description**. This will change the transformation to a Join.
- 3) Draw a line from **color** to **Description**. Your map should now look like this.



\_\_\_ k. Map **retailItems** to **ProductPackages**. This requires a submap.

- 1) Drag a line between **ClipItem []** and **Siebel\_ProductPackage []** to define a **submap** transform.



- 2) Click on **Submap**.

- 3) Click on the **Details** tab in the Properties view.
- 4) Click the **New...** button by the **Business Object Map** field.
- 5) Set the following values in the top portion of the dialog:
  - Module: SiebelAdapter
  - Namespace: (accept the default)
  - Folder: com/clipsandtacks/scm
  - Name: ClipItemToSiebelProductPackage
- 6) Click **Next**.
- 7) For both the **Input Type** and **Output Type**, click on the **Business Object** radio button so that the lists are filtered to only show Business Objects.
- 8) For the **Inputs**, select **ClipItem**.
- 9) For the **Outputs**, select **Siebel\_ProductPackage**.
- 10) Click **Finish**. This will put you into the map editor for the map you just created.

▼ Business object map

ClipItemToSiebelProductPackage

▼ Transformations

ClipItem	
itemID	string
GTIN	string
package	string
quantity	string
fullDescription	string
price	string
startDate	string
endDate	string
contactFirstName	string
contactLastName	string

Siebel_ProductPackage	
GlobalIdentifier	string
PackageDescription	string
Price	string
AvailabilityStart	string
AvailabilityEnd	string
Contact	string
version	token
delta	boolean
locale	string

- \_\_\_\_ 30. Map the fields from **ClipItem** to **Siebel\_ProductPackage**.
- \_\_ a. By using the click, hold and drag technique, define Move transformations for the following:
- GTIN > GlobalIdentifier
  - fullDescription > PackageDescription
  - price > Price
  - startDate > AvailabilityStart

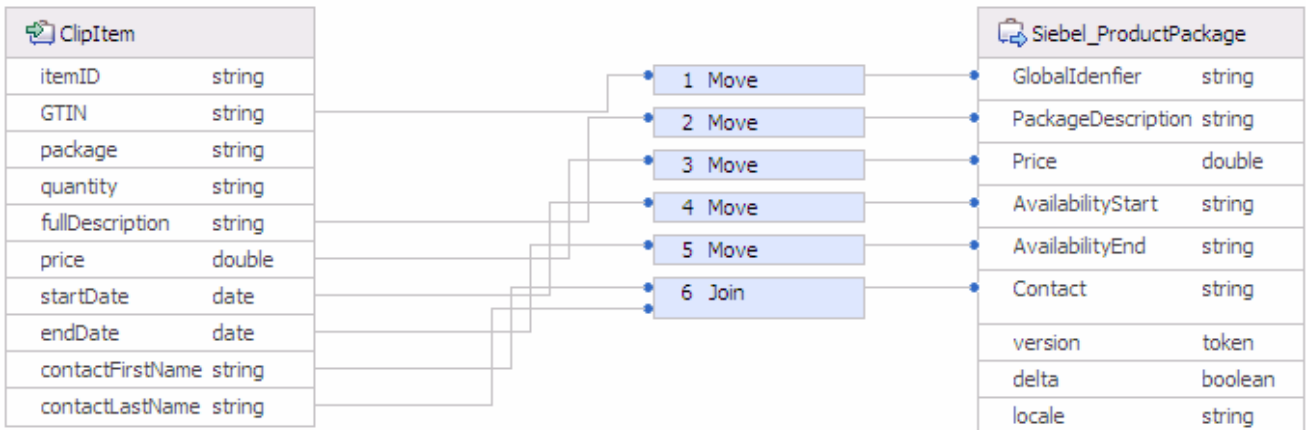
endDate > AvailabilityEnd

\_\_\_ b. The Contact field will contain the name in the firstname lastname format. Create a Join transformation to do this.

1) Draw a line from **contactFirstName** to **Contact**.

2) Draw a line from **contactLastName** to **Contact**.

\_\_\_ c. Your map should look like this:



\_\_\_ d. Press **Ctrl-S** to save the map.

\_\_\_ 31. The Interface Map and all three Data Maps are now complete. Prior to proceeding, ensure that all the maps you have been working with are saved.

\_\_\_ m. Examine the tabs at the top of the editors pane.



\_\_\_ n. For each one with an asterisk (\*), click on the tab to bring that editor to the foreground and then press **Ctrl-S** to save.

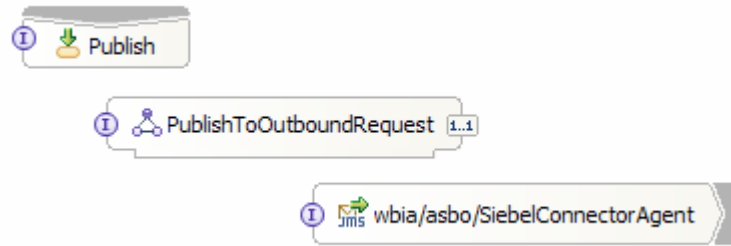
\_\_\_ 32. You are now ready to add the Interface Map to the wiring diagram.

\_\_\_ a. Click on the **Assembly Diagram: SiebelAdapter** tab on the top of the editors pane.

\_\_\_ b. Navigate to **SiebelAdapter > Mapping > Interface Maps** in the Business Integration view.

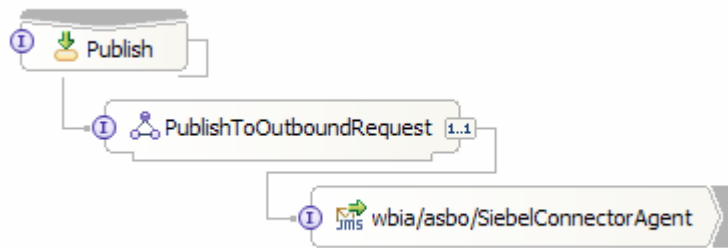
\_\_\_ c. Click on **PublishToOutboundRequest** and drag it onto the Assembly Editor.

\_\_\_ d. Rearrange the assembly diagram by dragging the elements so that they look like this:



\_\_\_ e. Draw a wire from the **Publish** Export to the **PublishToOutboundRequest** Interface Map.

\_\_\_ f. Draw a wire from the **PublishToOutboundRequest** Interface Map to the **SiebelConnectorAgent** Import.

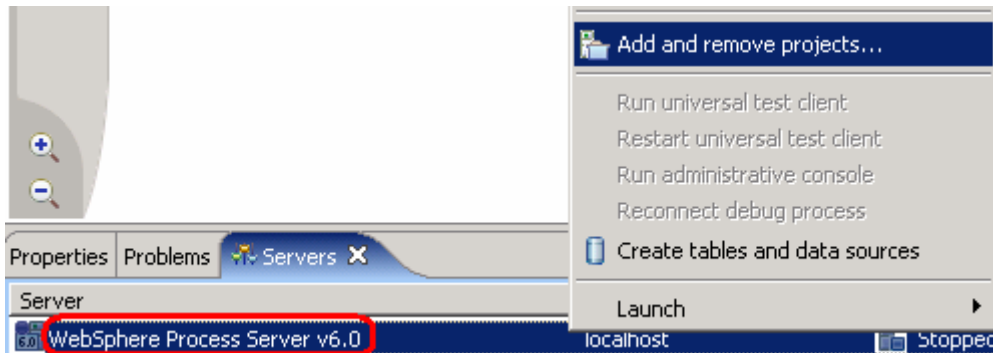


\_\_\_ g. Press **Ctrl-S** to save the Assembly Diagram.

## Part 4: Test the PeopleSoft Maps

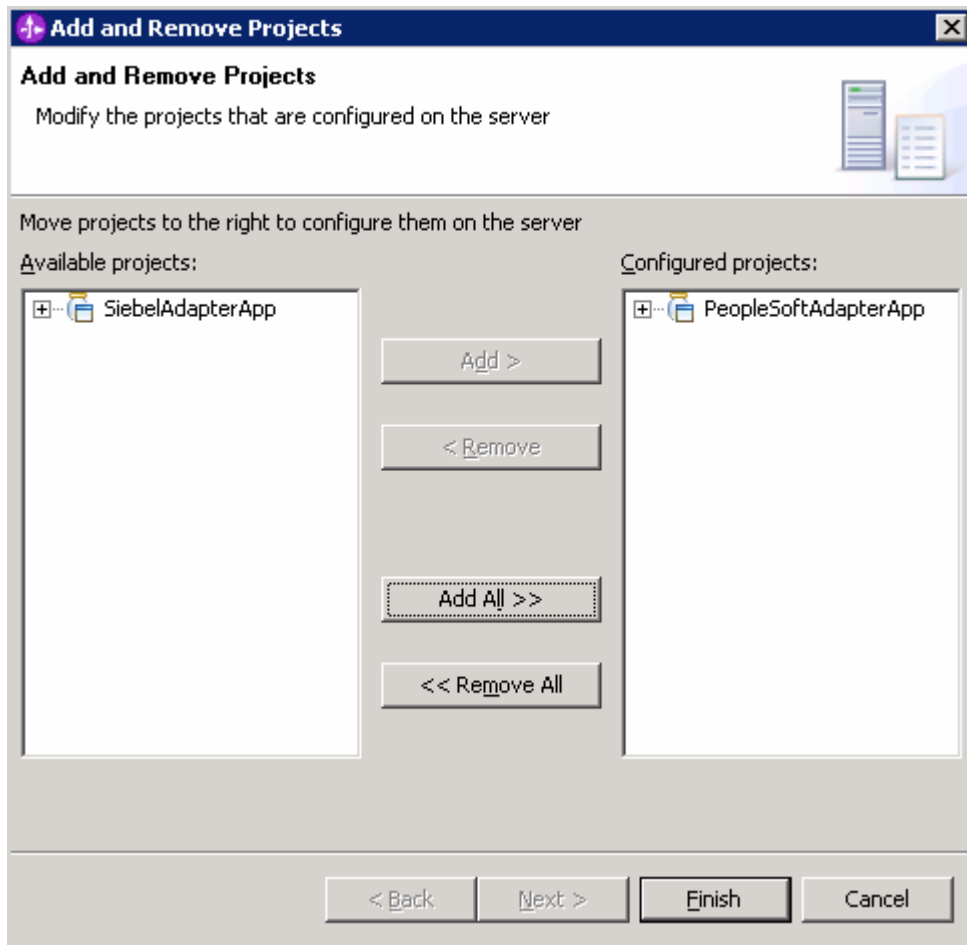
There are two ways that the maps created for PeopleSoft can be tested. The first approach would be to use the Test Component to drive the Interface Map on the input side and to see the result on the output side. This would be a unit test of just the maps. The other approach would be to use the Visual Test Connector to drive input through the Export which is wired to the Interface Map and then to use the Test Component to see the result on the output side of the Interface map. This not only tests the maps but also the wiring to ensure your module works as a whole. This is the approach you will take in this section.

- \_\_\_ 1. Add the PeopleSoftAdapter module to the UTE server so that it can be tested.
  - \_\_\_ a. From the Server view, right click on the server and select **Add and Remove Projects...** from the popup menu.



- \_\_\_ b. In the Add and Remove Projects dialog, select the **PeopleSoftAdapterApp** project from the Available projects panel.

\_\_\_ c. Click **Add >** to add it to the Configured projects panel.

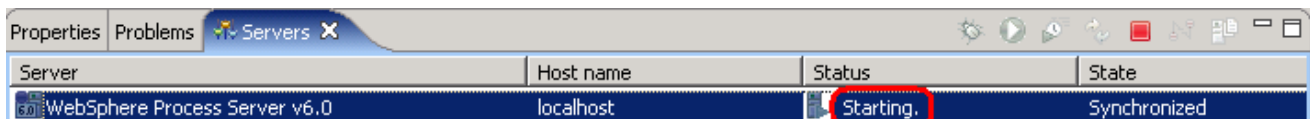


\_\_\_ d. Click **Finish**.

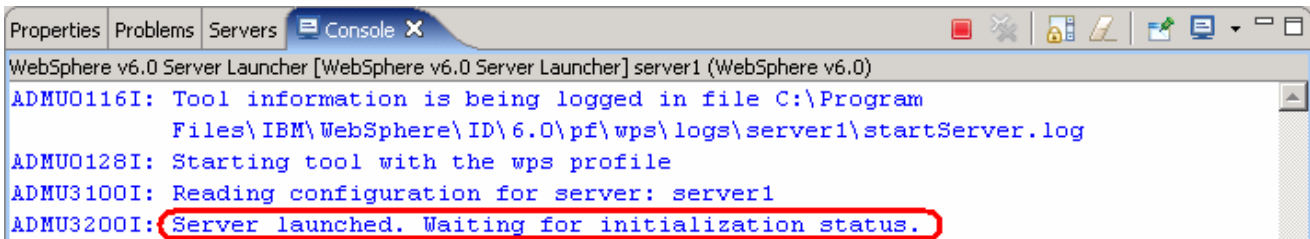
\_\_\_ e. If using a remote testing environment, follow the directions provided in [Task: Adding Remote Server to WebSphere Integration Developer Test Environment](#) (at the end of this document) to restart a server.

If using a local testing environment, the server will automatically begin to start so that the module can be added to the server.

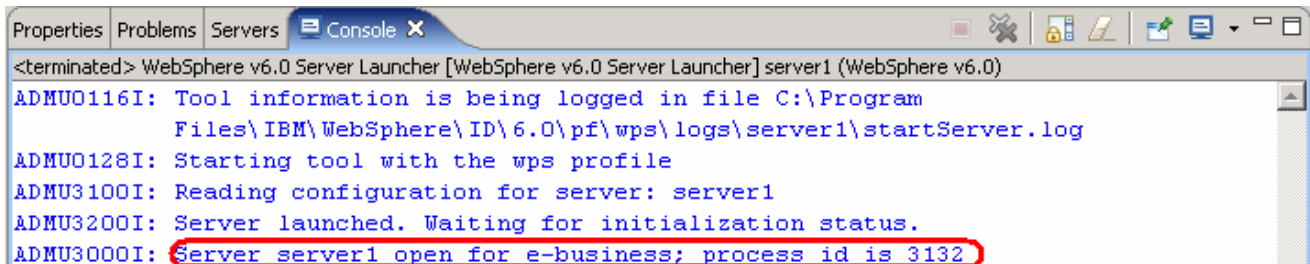
\_\_\_ f. You will at first see the following in the Servers view in the lower right panel.



\_\_\_ g. You will then see the Console view appear with something similar to the following:



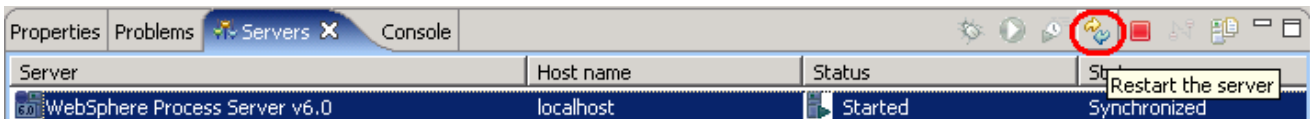
\_\_\_ h. Wait for the server startup to complete. This may take several minutes. Look for the “Server server1 open for e-business” message in the console.



\_\_\_ 2. The server must be restarted at this point for the messaging engine to be fully in initialized and ready to receive input from the Visual Test Connector.

\_\_\_ a. 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.

If using a local testing environment, click the **Restart the server** icon.



\_\_\_ b. Wait for the server restart to complete.

\_\_\_ 3. Connect the Visual Test Connector to the UTE Server.

\_\_\_ a. Using the connector configuration you created previously and saved at **<LAB\_FILES>\eXchange\PeopleSoft\ASBOs\PsoftConnectorWPS.cfg** follow the instructions below.

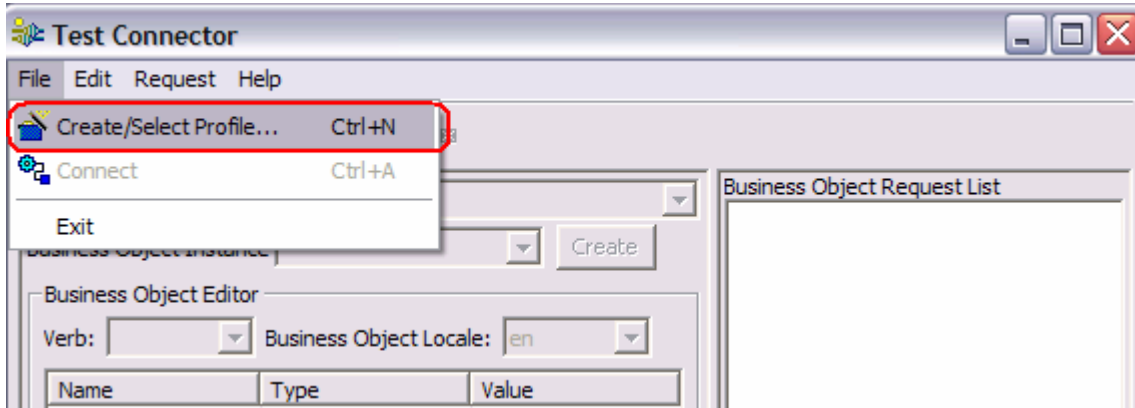
\_\_\_ 4. Ensure the WebSphere Process Server is active (the server is the JMS Broker). The VTC requires an active JMS Broker to start its connection.

\_\_\_ 5. Start the Visual Test Connector.

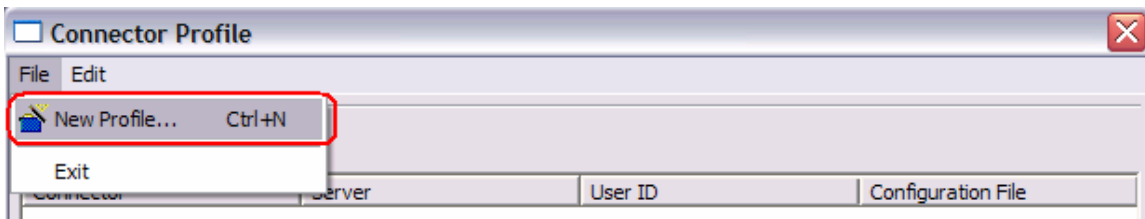
\_\_\_ a. From the menu, select **Programs > IBM WebSphere Business Integration Adapters > IBM WebSphere Business Integration Toolset > Visual Test Connector (VTC)**. This opens a command window in the background where the process runs and the Visual Test Connector Interface in the foreground. Do not close the command window as this will cause the VTC to exit.

\_\_\_ 6. Create/Select a Profile.

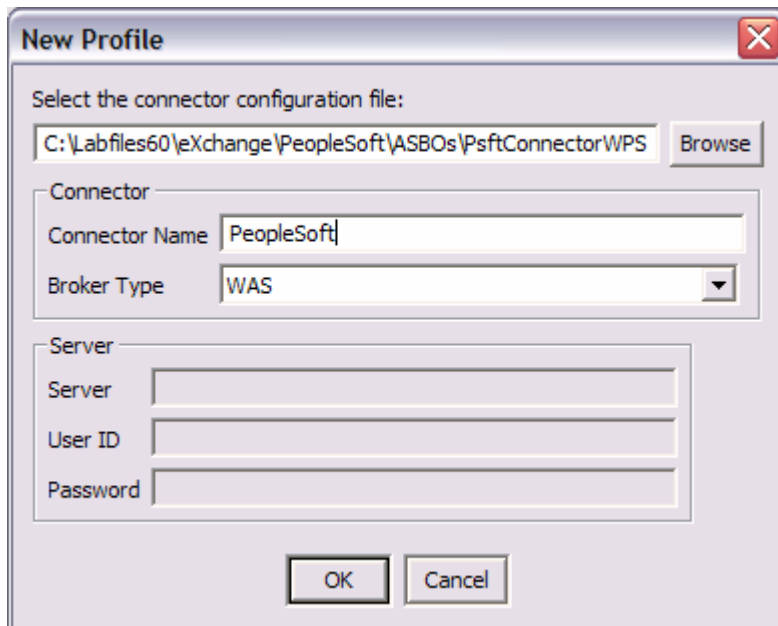
\_\_ b. From the VTC interface, select **File > Create/Select Profile**.



\_\_ c. If a profile does not already exist, create one by selecting **File > New Profile ...**. If a profile does already exist skip to step d.

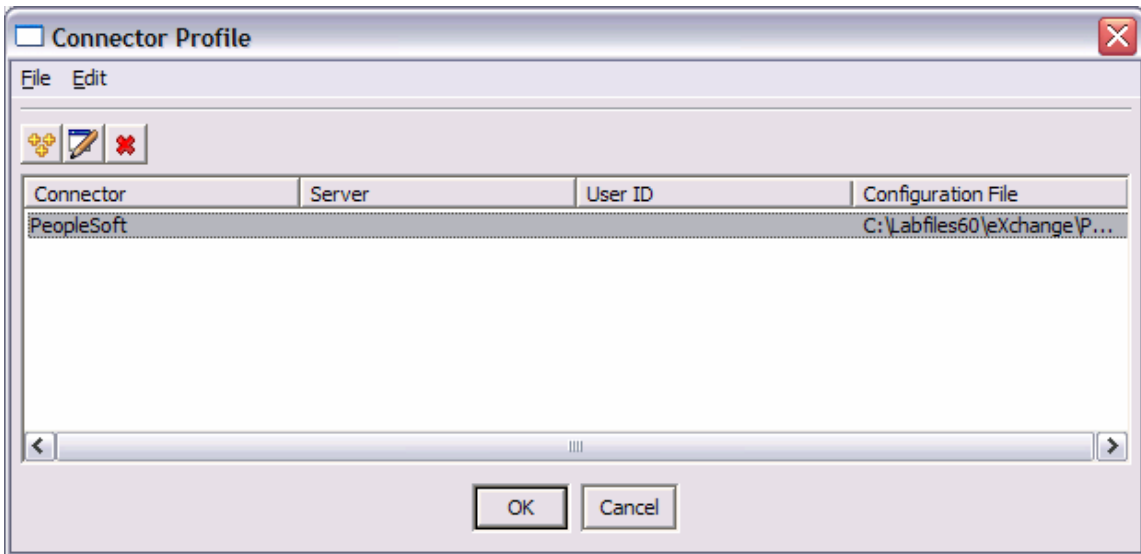


\_\_ d. Browse to and select the connector configuration file, type in a Connector Name for the Profile, and select WAS for the Broker Type. Click **OK**.

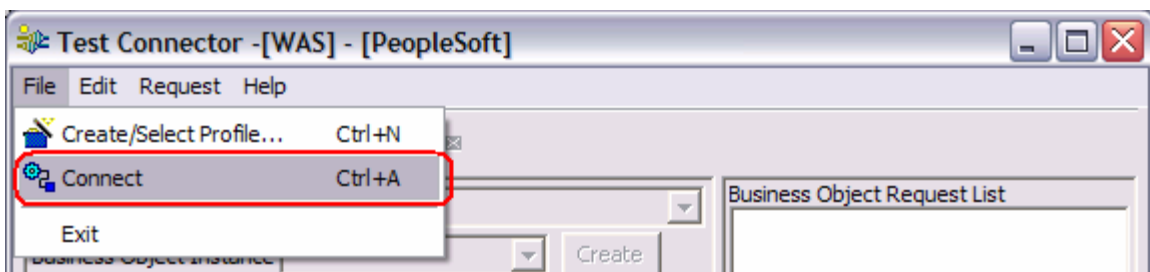




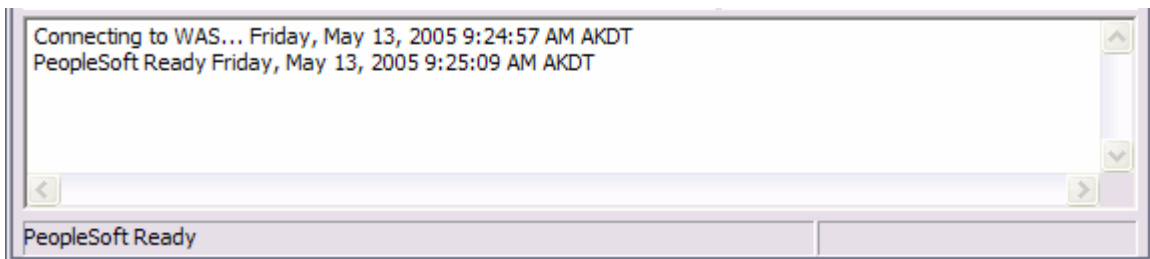
\_\_\_ e. Select and highlight the profile and click **OK**



\_\_\_ f. Select **File > Connect**.



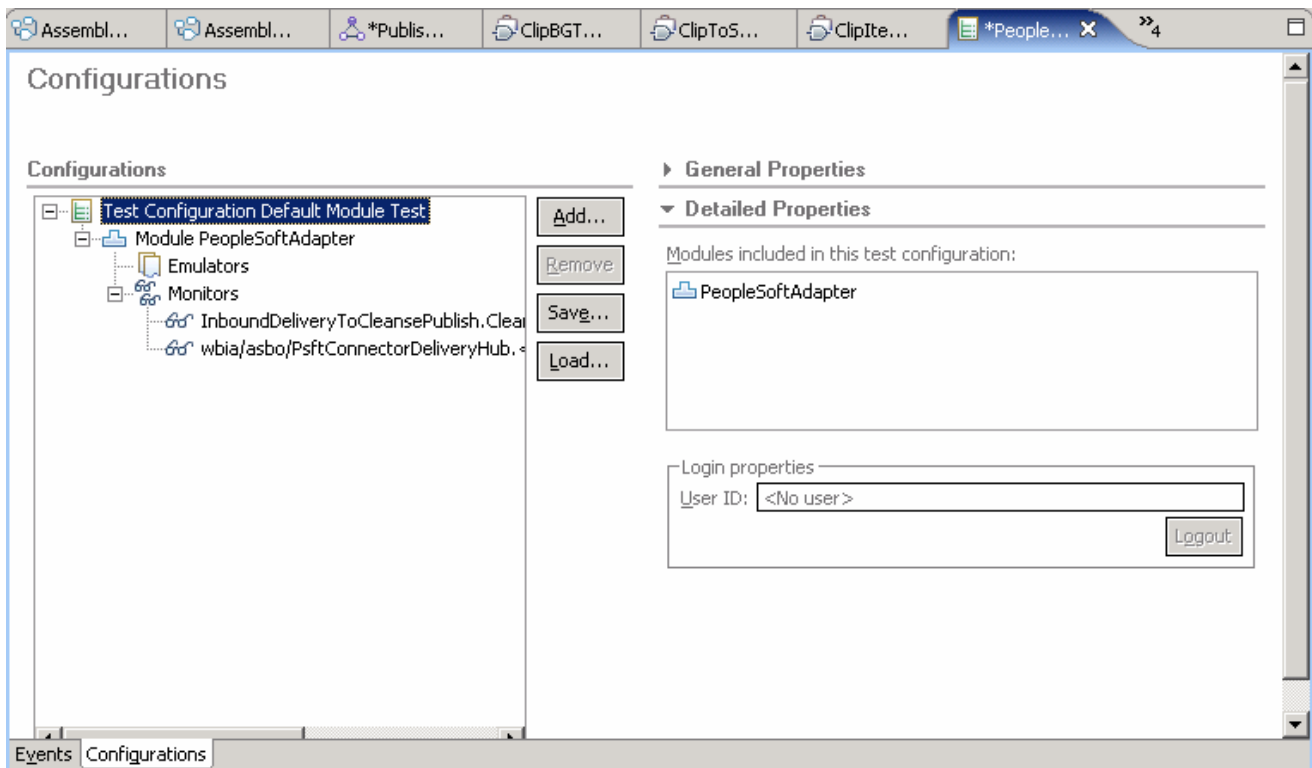
\_\_\_ g. View messages in the lower window.



\_\_\_ 7. Attach the test component to the PeopleSoftAdapter module.

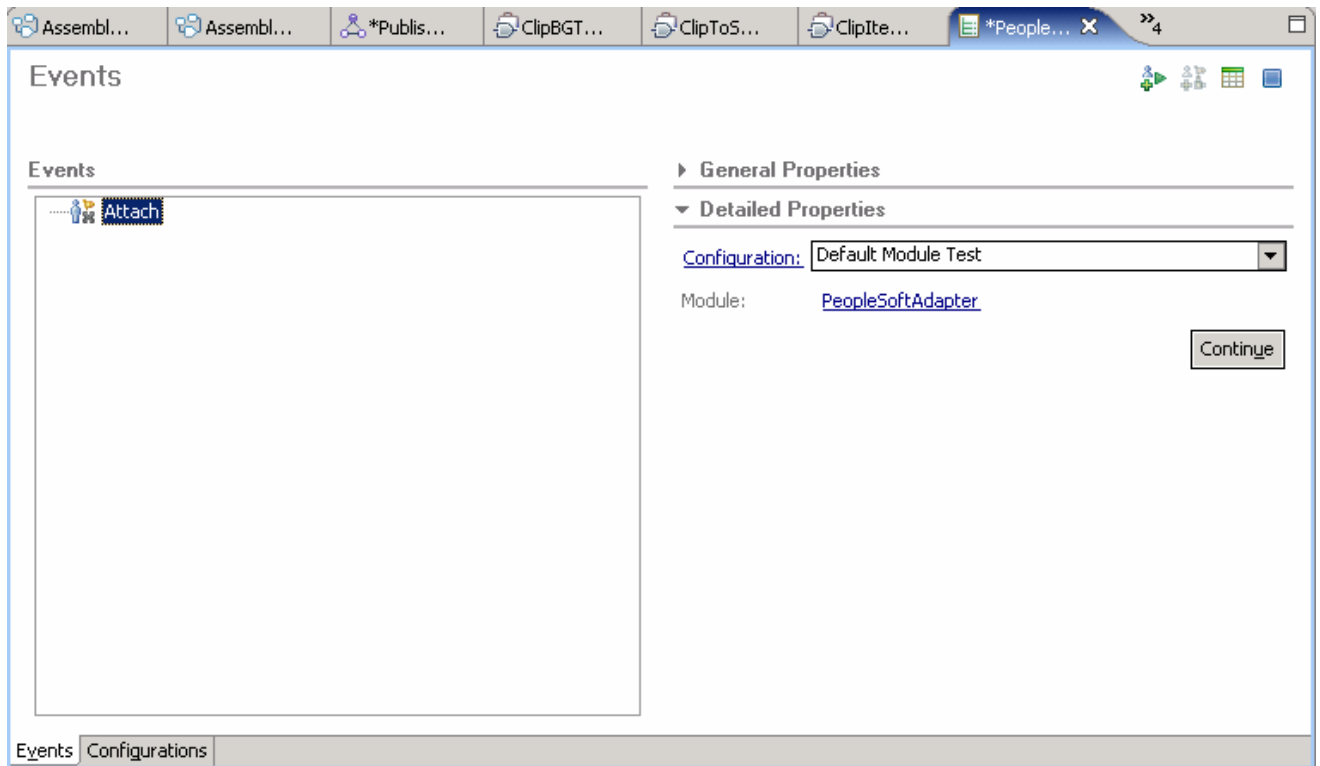
\_\_\_ a. In the Business Integration view, right click on the **PeopleSoftAdapter** and select **Test > Attach**.

- \_\_\_ b. Click on the **Configurations** tab at the bottom of the **PeopleSoftAdapter\_Test** view. You should see something like the following:

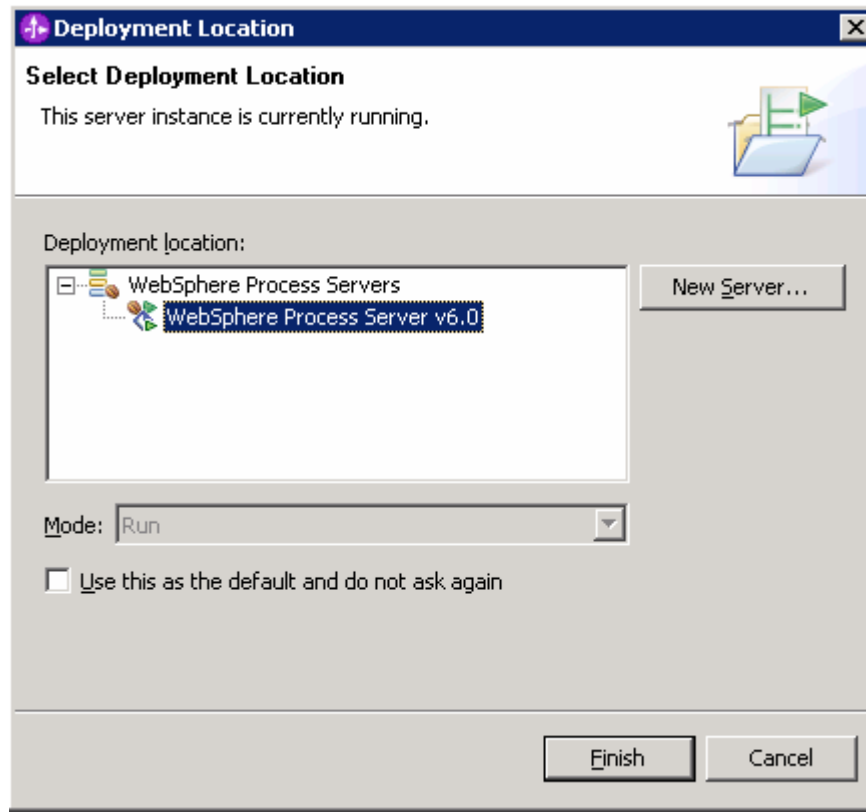


- \_\_\_ c. Notice **PsftConnectorDeliveryHub.<export> -> InboundDeliveryToCleansePublish** and **InboundDeliveryToCleansePublish -> CleansePublish** are listed under monitors. These are the wires, and the monitors will allow you to look at the data that passed through the wire when the test is run.

- \_\_\_ d. Click on the **Events** tab at the bottom of the **PeopleSoftAdapter\_Test** view. You should see something like the following screen capture. This is where the results will be shown when the application is invoked.



- \_\_\_ e. Click on **Continue**.
- \_\_\_ f. 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



\_\_\_ g. Click **Finish**. The Test Component is now ready to monitor the execution flow.

\_\_\_ 8. Load a predefined business object into the Visual Test Connector.

\_\_\_ a. In the VTC window, select **Edit > Load Business Object**.

\_\_\_ b. In the Open dialog, navigate to **c:\Labfiles60\exchange\PeopleSoft\ASBOInstances** and select **PsftASBO1.bo**.

\_\_\_ c. Click **Open**.

\_\_\_ d. In the New Instance dialog, enter a name to be assigned to this instance, such as **Psft1**.

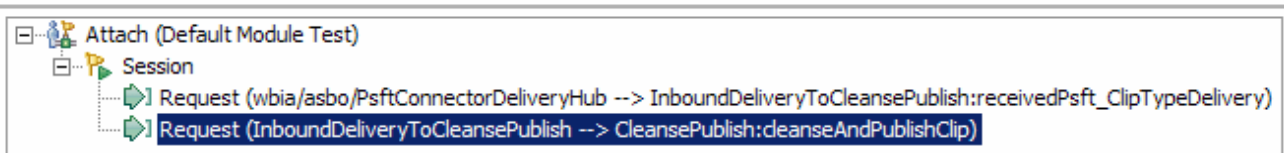
\_\_\_ e. Click **OK**.

- \_\_\_ f. This is the same ASBO that was used in the previous lab. You can examine its contents to be familiar with the data it will be sending into the server.

Name	Type	Value
ClipID	String	CID12345
Clip	String	clasp
Size	String	small
Color	String	black
[-] RetailUnits[n]	Psft_RetailItem	
[-] [0]	Psft_RetailItem	
ItemCode	String	98765
Package	String	box
Quantity	String	40
Price	String	2.59
Available...	String	2005-03-01
Withdraw...	String	2010-06-30
OwningC...	String	Jones, Sam
ObjectEv...	String	
ObjectEventId	String	

- \_\_\_ 9. Send the business object to the server.
  - \_\_\_ a. Select **Request > Send** in the Visual Test Connector.
- \_\_\_ 10. Examine the resulting business object in the server test component.
  - \_\_\_ a. In the Events panel, use the + to expand **Attach** and then **Session**. You should see the Request Monitor for the wire between the InboundDeliveryToCleansePublish Interface Map and the CleansePublish Import.

**Events**



- \_\_\_ b. Click on **Request Monitor (InboundDeliveryToCleansePublish -> CleansePublish)**. The object that went over this wire will be visible on the right side in the Request parameters box.

Input parameters

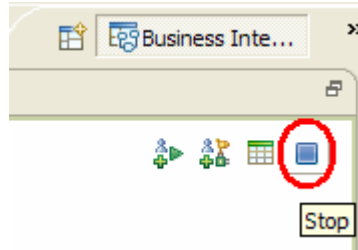
Name	Type	Value
[-] Psft_ClipTypeBGElement	Psft_ClipTypeBG	
verb	VerbType	Create
[-] Psft_ClipType	Psft_ClipType	
ClipID	ClipIDType	CID12345
Clip	ClipType	clasp
Size	SizeType	small
Color	ColorType	black
[-] RetailUnits	Psft_RetailItem [ ]	
[-] RetailUnits[0]	Psft_RetailItem	
ItemCode	ItemCodeType	98765
Packane	PackaneType	hnx

- \_\_\_ c. Scroll down to view the rest of the data.

\_\_\_ d. Compare the data from the Visual Test Connector with the data in the Test Component to verify that the mapping operations you defined worked properly.

\_\_\_ 11. Stop and close the test component.

\_\_\_ h. Click the **Stop** icon at the upper right corner Events pane.



\_\_\_ a. Click the **X** on the **PeopleSoftAdapter\_Test** tab.

\_\_\_ b. Click **No** in the Save Resource dialog.

\_\_\_ 12. Close the Visual Test Connector.

\_\_\_ a. Select **File > Exit** in the VTC window.

\_\_\_ b. Select **Yes** In the Shutdown dialog.

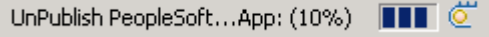
\_\_\_ c. Click the **enter** key in the Visual Test Connector command window to close it.

## Part 5: Test the Siebel Maps

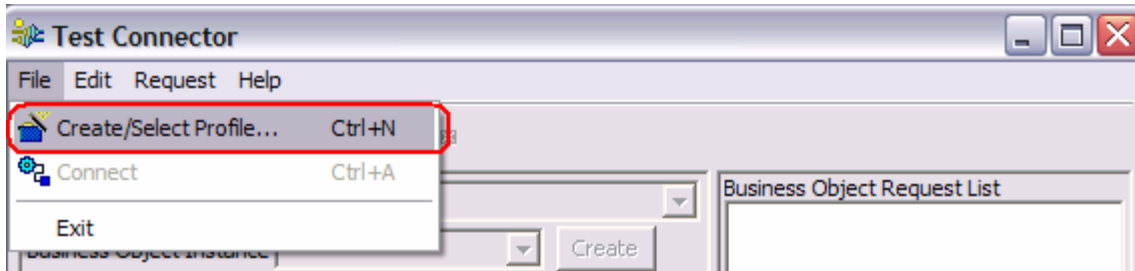
As with PeopleSoft, there are two ways that the maps created for Siebel can be tested. In both cases the Interface Map is driven with the Test Component. The first approach would be to also use the Test Component to monitor the output of the Interface Map and the other would be to let it flow all the way to the Visual Test Connector. The later is the approach you will take in this section.

This section assumes that the server is still running from the previous testing of the PeopleSoftAdapter module. However, these instructions should work whether the server is running or not when you begin.

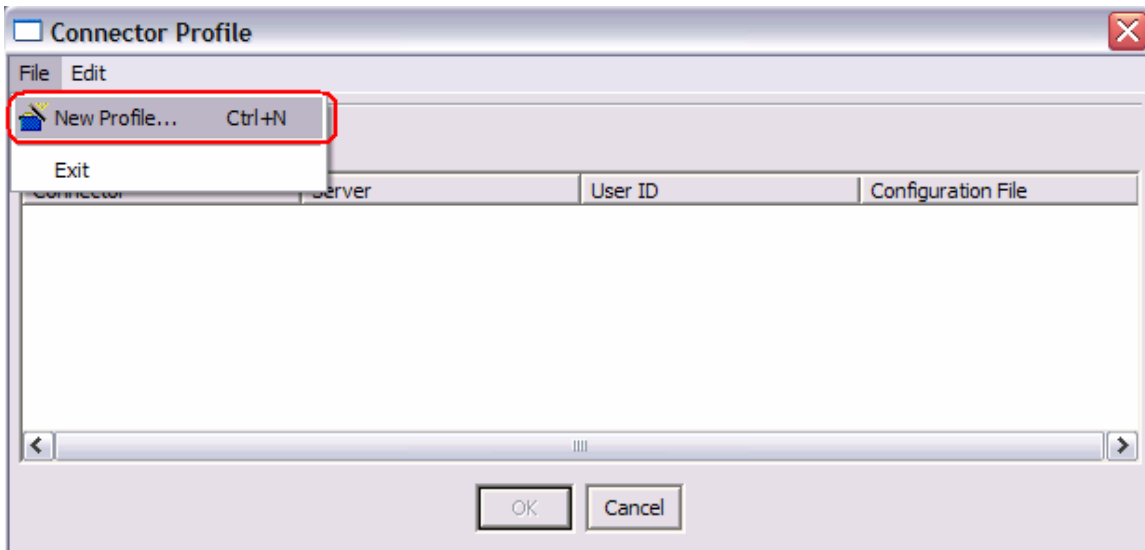
- \_\_\_ 1. Configure the UTE server so that the SiebelAdapter module can be tested.
  - \_\_\_ a. From the Server view, right click on the server and select **Add and Remove Projects...** from the popup menu.
  - \_\_\_ b. In the Add and Remove Projects dialog, use the **Add >** and **< Remove** buttons to configure the server so that the SiebelAdapter module is in the Configured projects list and that the PeopleSoftAdapter is in the Available projects list.
  - \_\_\_ c. Click on **Finish**.
  - \_\_\_ d. Wait for the server to complete the reconfiguration. Look in the lower right corner any you will see indicators such as this.



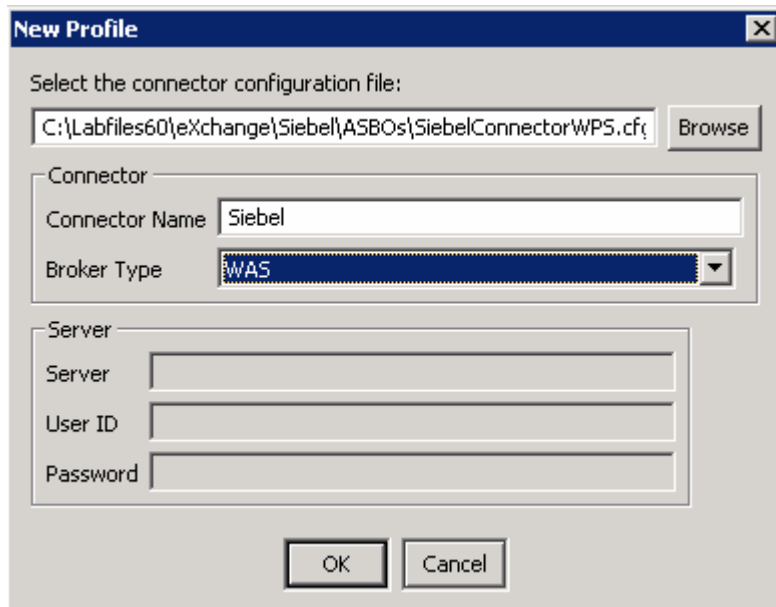
- \_\_\_ e. Wait for the publishing to finish. If your server was not started when you began this, there will be a wait while the server starts before publishing takes place.
- \_\_\_ 33. Connect the Visual Test Connector to the UTE Server.
  - \_\_\_ a. Using the connector configuration you created previously and saved at **<LAB\_FILES>\eXchange\Siebel\ASBOs\SiebelConnectorWPS.cfg** follow the instructions below.
- \_\_\_ 34. Ensure the WebSphere Process Server is active (the server is the JMS Broker). The VTC requires an active JMS Broker to start it's connection.
- \_\_\_ 35. Start the Visual Test Connector.
  - \_\_\_ i. From the **start** menu, select **Programs > IBM WebSphere Business Integration Adapters > IBM WebSphere Business Integration Toolset > Visual Test Connector (VTC)**. This opens a command window in the background where the process runs and the Visual Test Connector Interface in the foreground. Do not close the command window as this will cause the VTC to exit.
- \_\_\_ 36. Create/Select a Profile.
  - \_\_\_ j. From the VTC interface, select **File > Create/Select Profile**.



\_\_\_ k. If a profile does not already exist, create one by selecting **File > New Profile...** if a profile does already exist skip to step d.



\_\_\_ l. Browse to and select the connector configuration file, type in a Connector Name for the Profile, and select WAS for the Broker Type. Click OK.

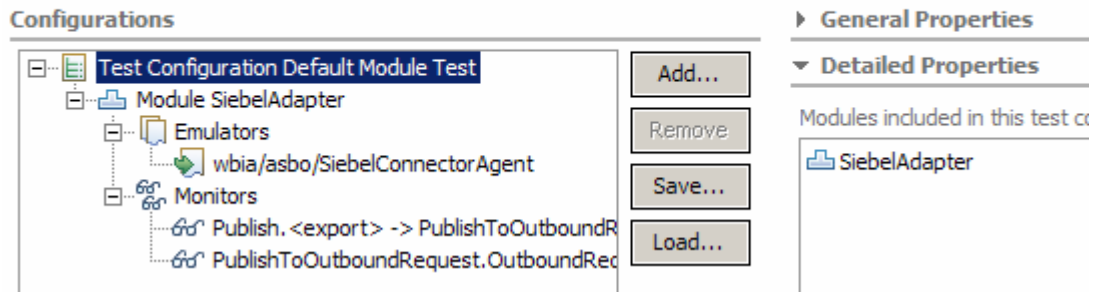






- \_\_\_ b. Click on the **Configurations** tab at the bottom of the **PublishToOubboundRequest\_Test** panel. You should see something like the following:

## Configurations



- \_\_\_ c. Notice that there is an Emulator defined for the SiebelConnectorAgent. This means the Test Component is set up to emulate the Import going to the Visual Test Connector rather than letting it process. Since you want to let this process, you need to remove this emulator.

- 1) Select the emulator **wbia/asbo/SiebelConnectorAgent**.
- 2) Click the **Remove** button.

- \_\_\_ d. Click on the **Events** tab at the bottom of the **PublishToOubboundRequest\_Test** panel.

- \_\_\_ e. Fill in the data values for the ClipBG.

For **verb**, use the drop down box in the value column to set the value **create**.

- \_\_\_ f. Fill in the data values for the Clip.

clipID: CID12345

GLN: 1234567890123

clip: clasp

size: small

color: black

brand: Clips and Tacks

- \_\_\_ g. Create a Clipltem instance.

- 1) Select and right click on **retailItems**.
- 2) Select **Add Element**.

- \_\_\_ h. Scroll down so that you can fill in the data values for the Clipltem.

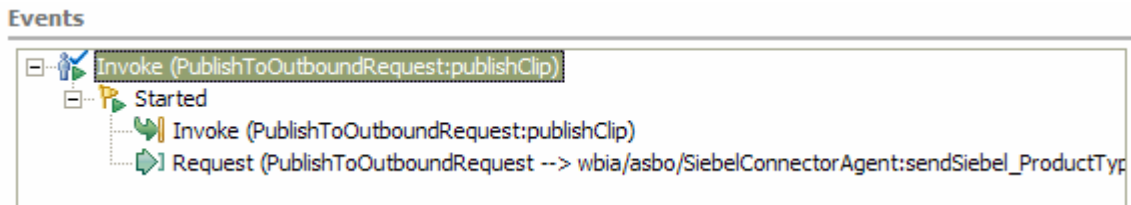
Set itemID: 98765.

Set GTIN: 12345678912345.

Set package: box.

- Set quantity: 40.
- Set fullDescription: box 40.
- Set price: 2.59.
- Set startDate: 2005-03-01.
- Set endDate: 2010-06-30.
- Set contactFirstName: Sam.
- Set contactLastName: Jones.

- \_\_\_ i. Save the object values so that you will not have to reenter them another time if you repeat the testing scenario.
  - 1) Scroll back to the top of the **Initial request parameters** panel.
  - 2) Select and right click on **ClipBG**.
  - 3) Select **Add value to Pool...**
  - 4) Give it a name, such as **inClipBG**.
  - 5) Click **OK**.
- \_\_\_ 39. Run the test.
- \_\_\_ p. Click **Continue**.
- \_\_\_ q. 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 FinishClick **Finish**.
- \_\_\_ 40. You can now examine the following items:
  - \_\_\_ r. Expand **Invoke > Started** in the Events panel and you should see the following. It shows the invocation on the PublishToOuboundRequest interface map and also the request which is what went over the wire to the SiebelConnectorAgent import.

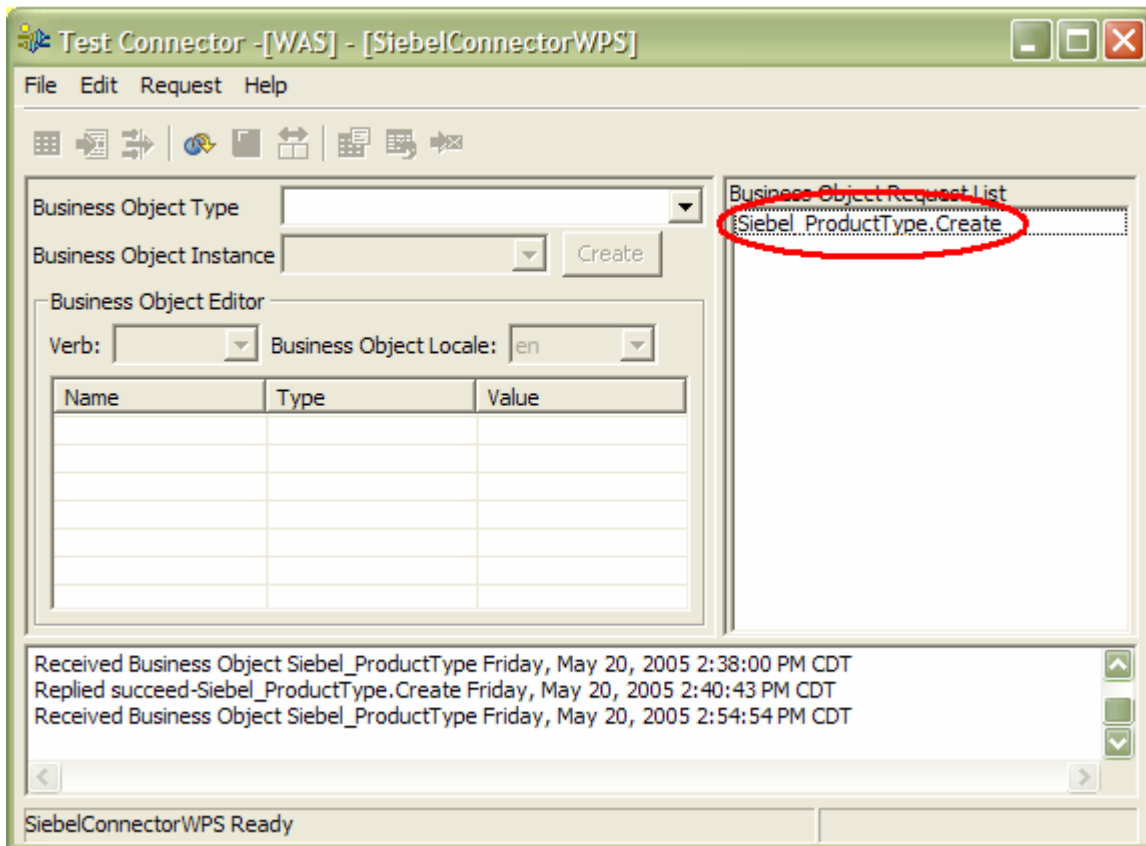


\_\_\_ s. Double click on the Request so that you can see the output of the Interface map.

Request parameters

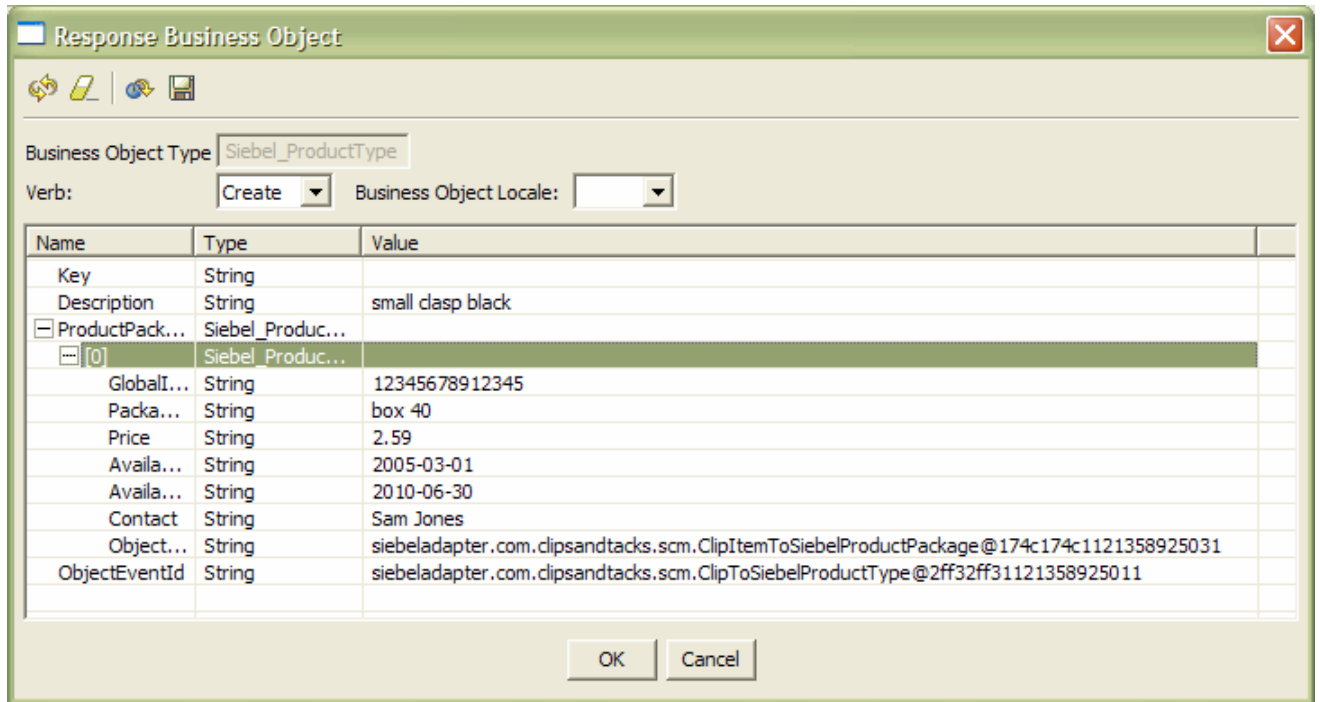
Name	Type	Value
[-] Siebel_ProductTypeBGElement	Siebel_ProductTypeBG	
verb	VerbType	Create
[-] Siebel_ProductType	Siebel_ProductType	
Key	KeyType	<null>
Description	DescriptionType	small clasp black
[-] ProductPackages	Siebel_ProductPackage [ ]	
[-] ProductPackages[0]	Siebel_ProductPackage	
GlobalIdentifier	GlobalIdentifierType	12345678912345
PackageDescription	PackageDescriptionType	box 40
Price	PriceType	2.59
AvailabilityStart	AvailabilityStartType	2005-03-01

\_\_\_ t. Switch to the Visual Test Connector window. You should see an entry in the Business Object Request List as shown below.



\_\_\_ u. Double click on the **Siebel\_ProductType.Create** and you will get a Response Business Object window showing you what was sent to the Visual Test Connector.

\_\_\_ v. Enlarge the window and expand the ProductPackages[n] and [0] elements. You should see something like the following:



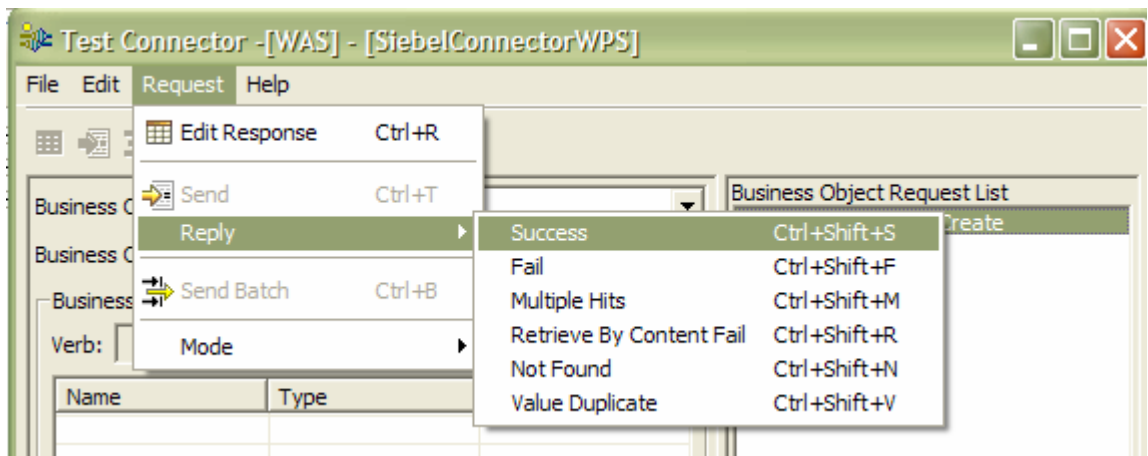
\_\_\_ w. Compare the values reported above with those entered in the Test Component to see if the mapping definitions you defined appear to have worked correctly. Also, these should be the same as what was passed on the wire.

\_\_\_ 41. Now you can send back a response to the server.

\_\_\_ x. In the Response Business Object window you will enter a value for the **Key = A1234**.

\_\_\_ y. Click OK.

\_\_\_ z. Send the response back to the server by selecting **Request > Reply > Success**.

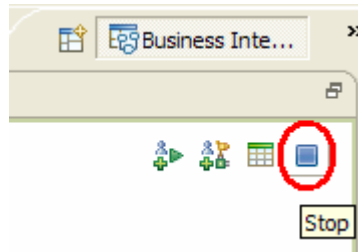


- \_\_\_ aa. You will see in the Test Component a Response to the Invocation. Double click on the Response so that the Response parameters show on the right panel. Examine the returned ebel\_ProductType business object to see the Key has been set to A1234.

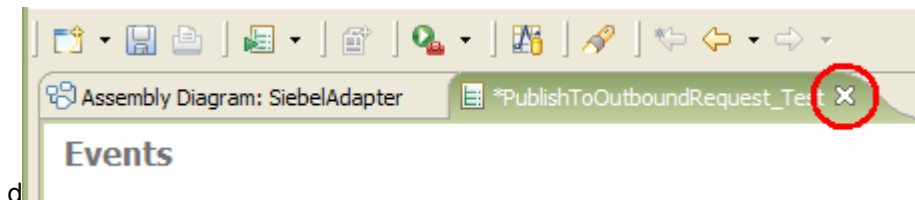
Response parameters

Name	Type	Value
[-] Siebel_ProductTypeBGElement	Siebel_ProductTypeBG	
verb	VerbType	Create
[-] Siebel_ProductType	Siebel_ProductType	
Key	KeyType	A1234
Description	DescriptionType	small clasp black
[-] ProductPackages	Siebel_ProductPackage [ ]	
[-] ProductPackages[0]	Siebel_ProductPackage	
GlobalIdentifier	GlobalIdentifierType	12345678912345
PackageDescription	PackageDescriptionType	box 40
Price	PriceType	2.59
AvailabilityStart	AvailabilityStartType	2005-03-01

- \_\_\_ 42. Stop and close the test component.
- \_\_\_ bb. Click the **Stop** icon at the upper right corner Events pane.



- \_\_\_ cc. Click the **X** on the **PublishToOutboundRequest\_Test** tab.



- \_\_\_ dd. Click **No** in the Save Resource dialog.
- \_\_\_ 43. Close the Visual Test Connector.
- \_\_\_ a. Select **File > Exit** in the VTC window.
- \_\_\_ b. Select **Yes** in the Shutdown dialog.
- \_\_\_ 44. Remove the SiebelAdapter from the server.
- \_\_\_ a. Right click on the server from the Servers view and select **Add and Remove Projects...** from the popup menu.
- \_\_\_ b. Select **SiebelAdapter** and click the **< Remove** button in the Add and Remove Projects dialog.

- \_\_\_ c. Click on **Finish**.
- \_\_\_ d. Wait for the server to complete the reconfiguration.
- \_\_\_ 45. Stop the Server
- \_\_\_ a. Right click on WebSphere Process Server v6.0 server from the Servers view and select Stop from the context menu.

## **What you did in this exercise**

You defined Interface Maps and Data Maps which were used to map between ASBOs and GBOs. You tested these maps using the Component Test capabilities in WebSphere Integration Developer.



## Solution Instructions

- \_\_\_ 2. Follow the directions in the task [Initialize the Workspace for a Lab Exercise](#), using the following values:

**<WORKSPACE>**

C:\Labfiles60\exchange\WBIAdapters\workspaceMaps

**<PROJECT\_INTERCHANGE>**

C:\Labfiles60\exchange\WBIAdapters\Solution\Mapping\_PI.zip

**<MODULE>**

n/a

**<DEPENDENT\_LIBRARIES>**

n/a

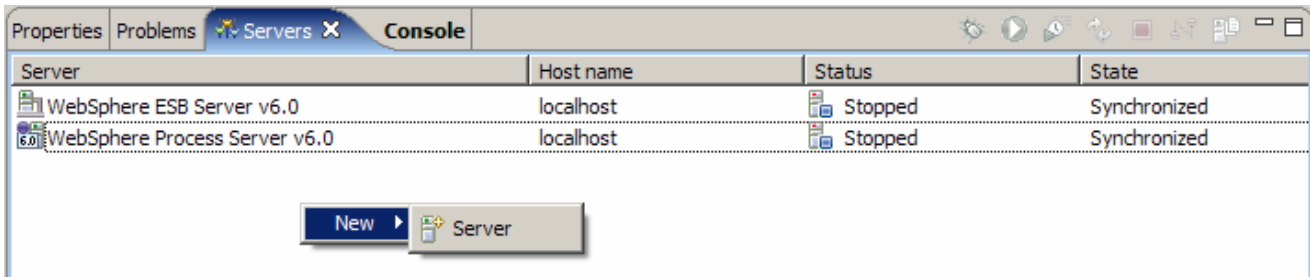
- \_\_\_ 46. Continue with Part 4: **Test the PeopleSoft Maps.**

## Task: Adding Remote Server to WebSphere Integration Developer Test Environment

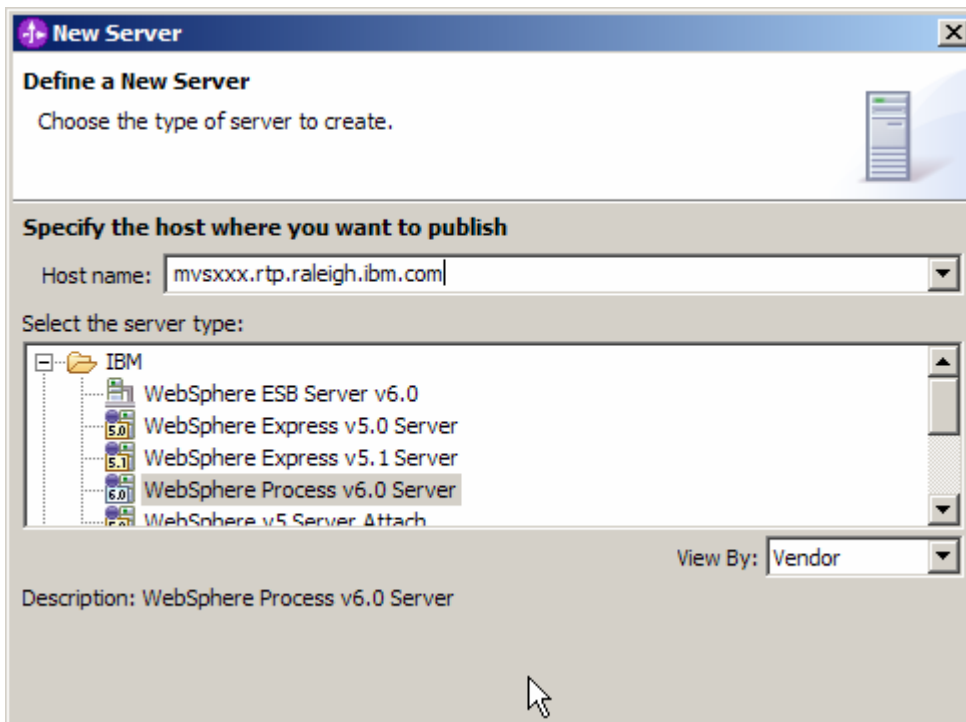
This task describes how to add a remote server to the WebSphere Integration Developer Test environment. The sample will use a z/OS machine.

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>**).

**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: [ server1 ]

Server type

BASE, Express or unmanaged Network Deployment server

Network Deployment server

Network Deployment server name: [ ]

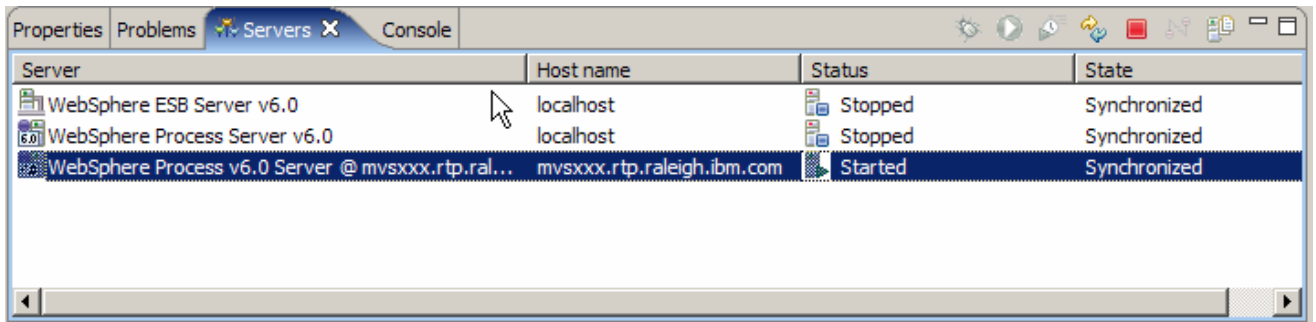
The server name is in the form of:  
<cell name>/<node name>/<server name>  
For example, localhost/localhost/server1.

Click this button to detect the server type.

< Back   Next >   Finish   Cancel

\_\_\_ 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.

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

**'telnet <HOSTNAME> <TELNET\_PORT>'**

userid : <USERID>

pw : <PASSWORD>

\_\_ b. Navigate to the bin directory for the profile being used:

**cd <WAS\_HOME>/profiles/<PROFILE\_NAME>/bin**

\_\_ c. Run the command file to start the server: **./startServer.sh <SERVER\_NAME>**

\_\_ d. 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
    
```