IBM WebSphere® Enterprise Service Bus V6.0.2 – Lab exercise

WebSphere Enterprise Service Bus lab – Dynamic endpoints

What this exercise is about

The objective of this lab is to provide an understanding of how to create a Mediation Flow that will set a dynamic endpoint address in a Service Message Object (SMO) and invoke a service using the address.

Lab requirements

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

- WebSphere Integration Developer V6.0.2 with the WebSphere Enterprise Service Bus test server option installed

What you should be able to do

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

- Import the project interchange file into the WebSphere Integration Developer V6.0.2 development environment
- Create and edit a mediation module and mediation flow
- Navigate the Properties View for mediation information
- Generate implementation and binding from the development environment
- Work with the Mediation Flow Editor and build XSL Transformation request and response flows
- Test by running a JSP on the WebSphere Enterprise Service Bus V6.0 server with the Dynamic End Point property enabled (default) for the Callout
- Test by running a JSP on the WebSphere Enterprise Service Bus V6.0 server with the Dynamic End Point property disabled for the Callout

Introduction

The concept of Dynamic End Points provide the runtime support, in fact it provides an enhancement to the WebSphere Enterprise Service Bus (and WebSphere Process Server) runtime to:

- Allow the mediation programming model to select (or to influence the selection of) service endpoints at the runtime
- Allow the selection at runtime of a service endpoint that has not been predefined in the mediation flow

With the support provided by Dynamic End Points, the mediation flow designer may use a Callout Node in the flow in a dynamic way. For a dynamic callout, the associated reference provided on the component need not be wired to an import. At runtime, the dynamic callout retrieves an element from the Service Message Object (SMO) which provides the endpoint Address to be used by the callout as the service endpoint.

The mediation flow primitives that are wired in the flow may provide logic which affects the contents of the endpoint Address held in the SMO, and the logic which may control the selection of the callout to be used. For example, a Database Lookup primitive may supply an Endpoint Reference (EPR) used to set endpoint information into the transient context of the SMO, and a Message Filter primitive may then route the request into the transient context of the SMO, and a Message Filter may then route the request (based on the characteristics of the endpoint information) through XSLT Transformation primitives which may tailor the final endpoint Address set in the request SMO. The SMO is then passed by way of a dynamic callout to invoke the required external service.

The runtime implementation of Dynamic End Points is based on the SCA runtime support for Dynamic References. The support offered by Dynamic End Points is therefore consistent with SCA Dynamic Reference support and is subject to the same limitations. For example, there is no direct support for dynamic invocation of a service using JMS Bindings, although a predefined SCA import which specifies JMS Bindings may be selected at runtime.

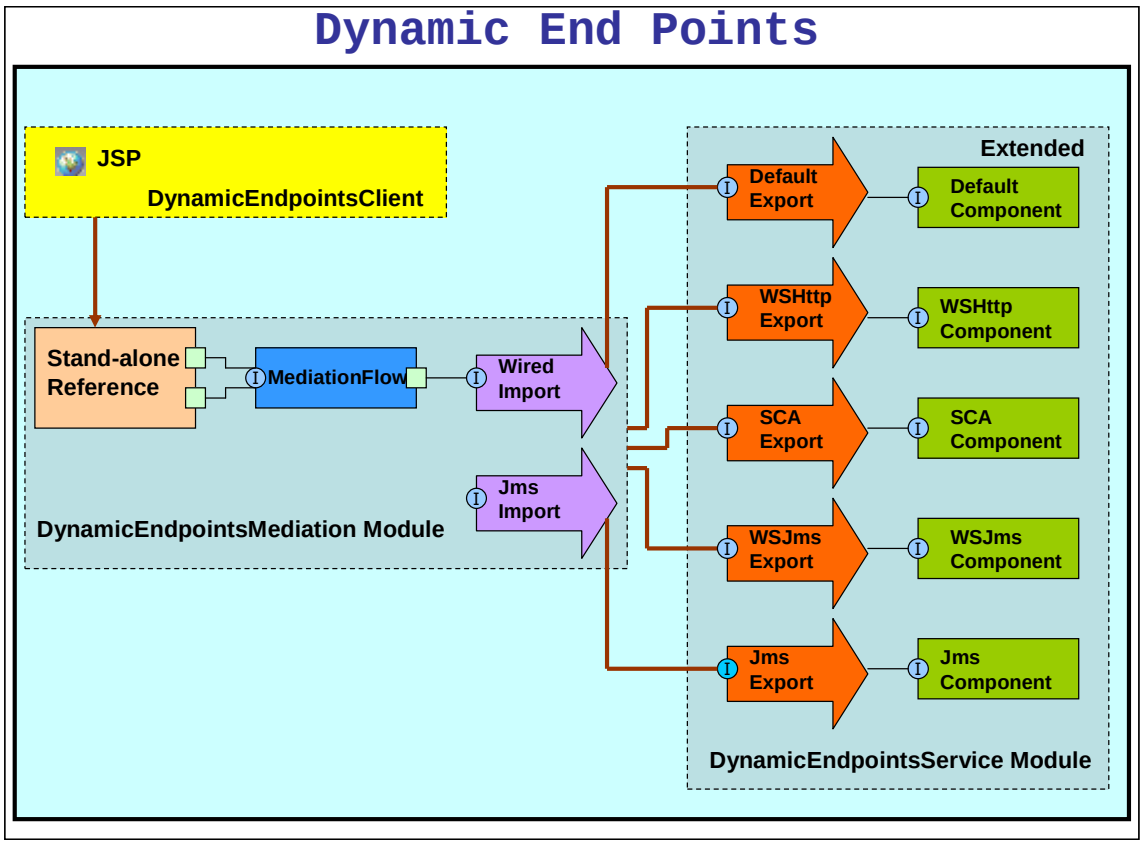
For use of the Dynamic End Points, as endpoint Address representation is carried in the Service Message Object (SMO) that is passed to the Callout node in a mediation flow. A new property of a Callout node controls whether the Address held in the SMO header may be used to dynamically invoke the service.

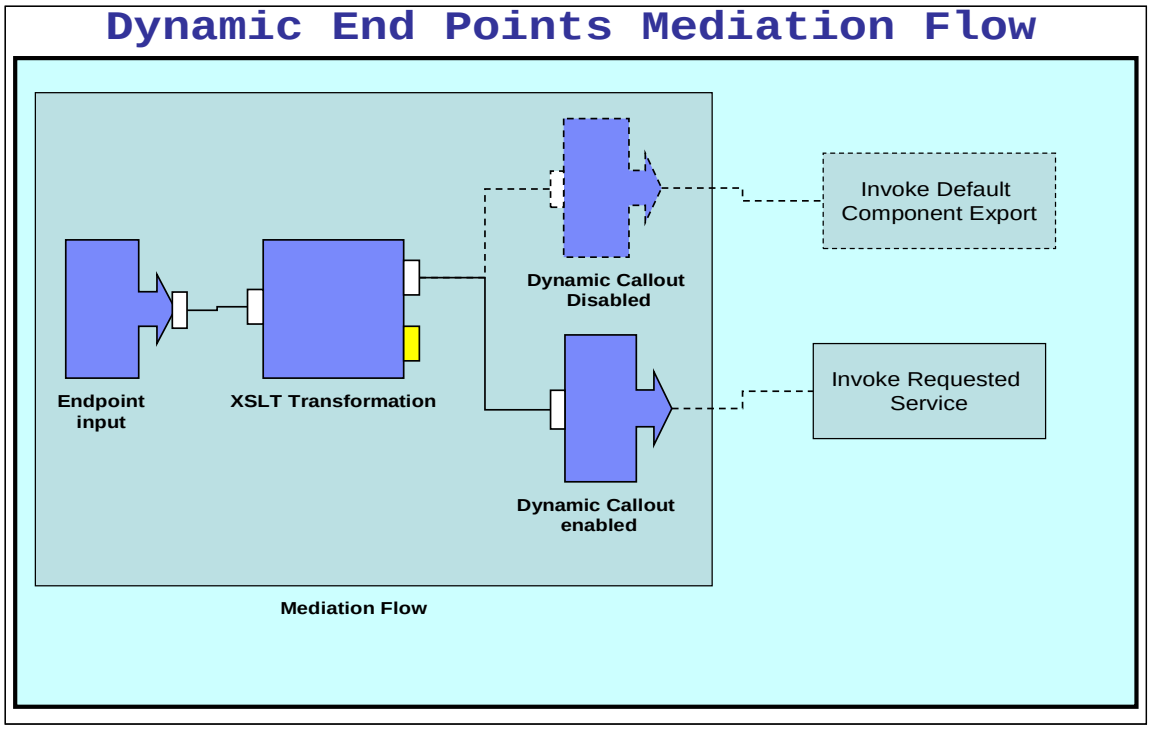
If the Callout property specifies that the dynamic Address may be used, and the SMO header at runtime holds an endpoint Address, then this Address is used by the runtime when invoking the service. The default endpoint (if any) is overridden by the Address located in the header of the SMO.

In this lab, you construct a Mediation Flow that includes a Callout to a service. Within the body of the message received by the Mediation Flow, there is an endpoint address. An XSLT primitive in the Mediation Flow moves the endpoint address from the SMO body to the dynamic endpoint address field in the SMO header.

The actual endpoint used by the Callout is either:

- The dynamic endpoint address in the SMO header (if Callout property "**Use dynamic endpoint**' enabled)
- The endpoint specified by the wired import (if Callout property "**Use dynamic endpoint**' disabled)





Exercise instructions

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

Reference Variable	Windows Location	AIX®/UNIX® Location
<WID_HOME>	Ex: C:\WID602	
<LAB_FILES>	C:\Labfiles602	/tmp/Labfiles602
<TEMP>	C:\temp	/tmp

Windows users' note: When directory locations are passed as parameters to a Java™ program such as EJBDeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles602\ would be replaced by C:/LabFiles602/

Note that the previous table is relative to where you are running WebSphere Integration Developer. This table is related to where you are running remote test environment:

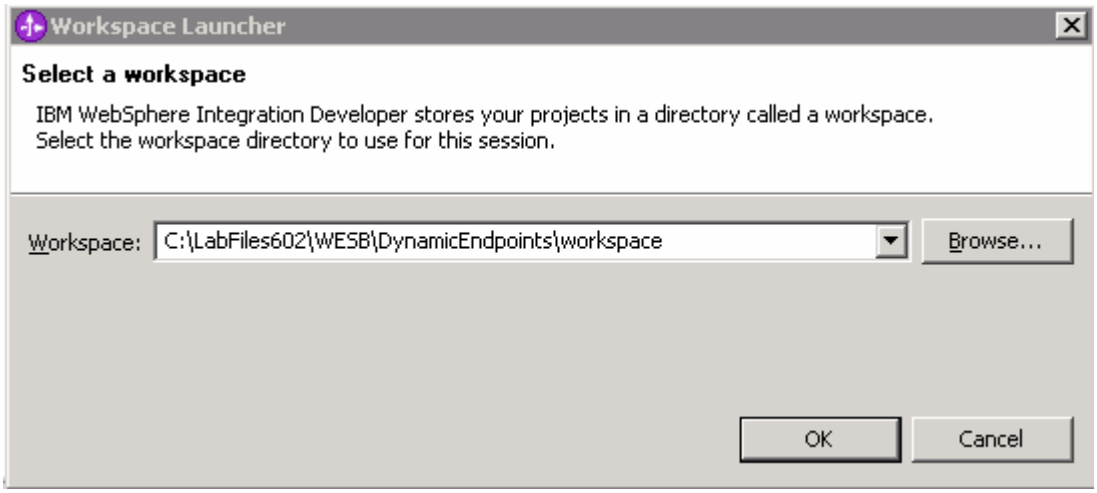
Reference Variable	Example: Remote Windows test server location	Example: Remote z/OS test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	cl1sr01	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\App Server	/etc/cl1cell/AppServerNode1	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<PORT>	9080	9080	
<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: Prepare environment for the lab

In this section of the lab, all the projects that are part of **WESB_DynamicEndpoints_PI.zip** project interchange file are imported into a new workspace. Remember this is the sample SCA application to which the ESB specific mediation is added as the lab progresses.

____ 1. Start WebSphere Integration Developer V6.0.2 with a workspace location of **C:\LabFiles602\WESB\DynamicEndpoints\workspace**



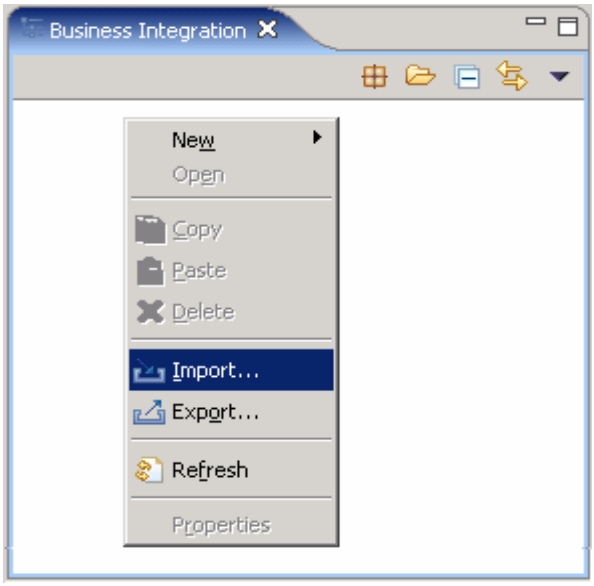
____ 2. On the welcome screen, click the curved arrow at the top right to "**Go to the business integration perspective** (),to close the Welcome screen.

NOTE: If this is not the first time WebSphere Integration Developer is open, you will not see this welcome screen. The welcome screen is only seen the first time WebSphere Integration Developer is open.

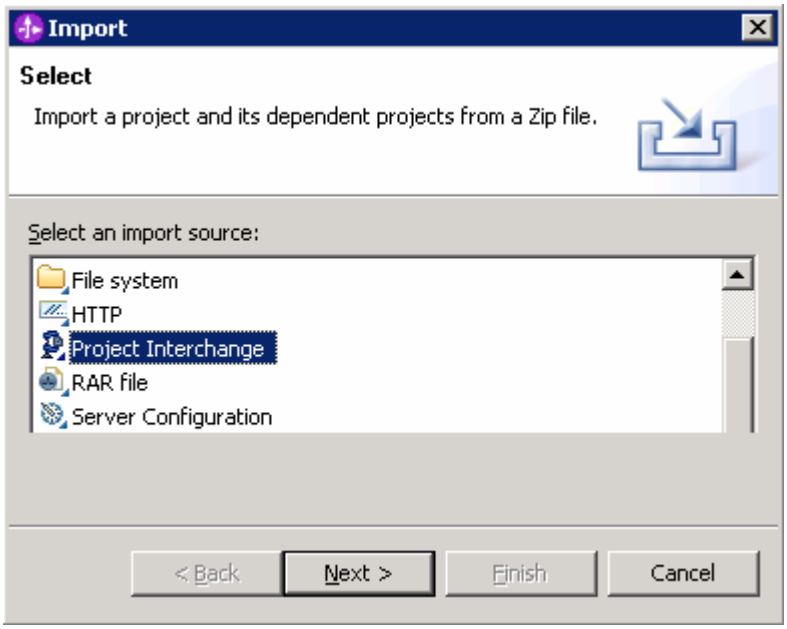
____ 3. Import the Project Interchange file, **WESB_DynamicEndpoints_PI.zip**, into the development environment

__ a. Right-click inside **Business Integration View** (top left view in the Business Integration Perspective)

__ b. Select **Import** from the context menu

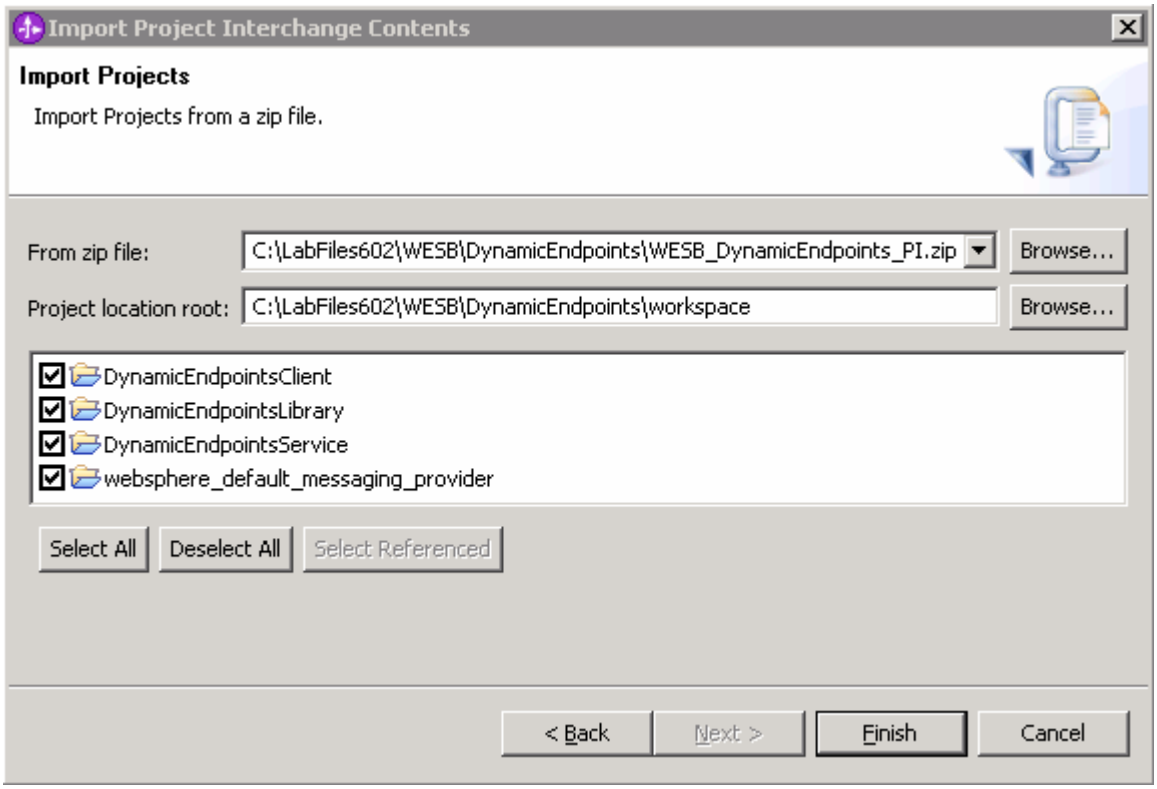


__ c. From the **Import** dialog, select **Project Interchange** from the list



__ d. Click **Next**

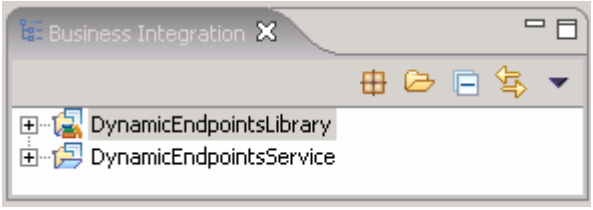
__ e. In the following screen, click the **Browse** button for "**From zip file"** to navigate for the Project interchange file, **WESB_DynamicEndpoints_PI.zip**



__ f. Click the **Select All** button to ensure all projects listed are selected

__ g. Click the **Finish** button (projects will be imported and auto-build will run)

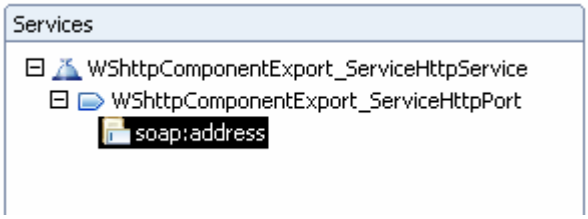
__ h. Verify that the **DynamicEndpointsLibrary** and **DynamicEndpointsService** modules are listed in the Business Integration view.



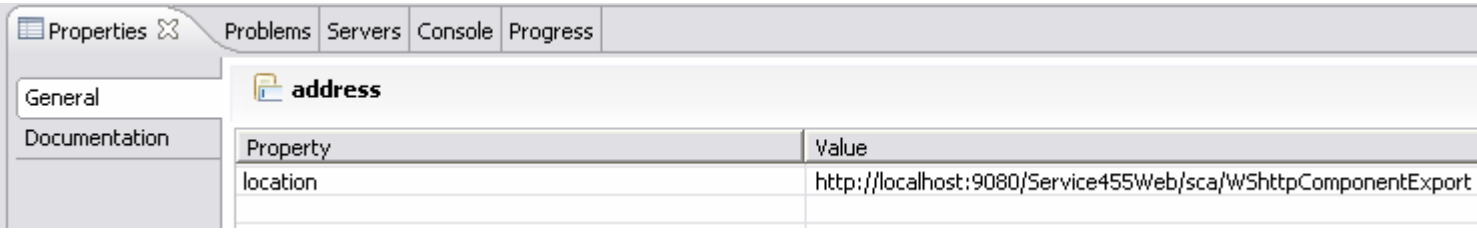
__ i. If <HOSTNAME>:<PORT> is something other than localhost:9080, change the address in the export as shown here.

1) From the **Business Integration Perspective**, select **WShttpComponentExport_ServiceHttpPort** from the **Web Service Ports** section**.** You will then be in the WDSL editor.

2) From the WSDL editor, expand the **WShttpComponentExport_ServiceHttpService** service until you see the **soap:address** selection. Select that to open the properties view in the lower pane.

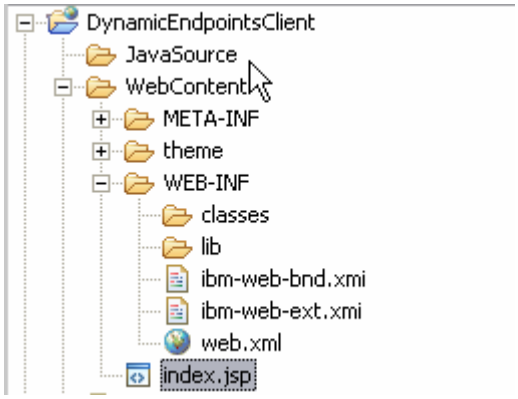


3) Change the **localhost:9080** found there to **<HOSTNAME>:<PORT>** (for example, MVS222.rtp.raleigh.ibm.com:9138) to match your remote system.



4) **Ctrl+S** to save the changes and close the WSDL Editor

5) From the **Physical Resources** perspective, select **index.jsp** under **DynamicEndpointsClient.**

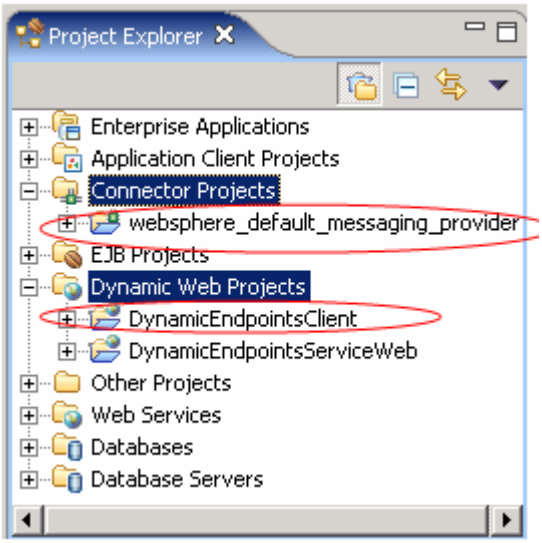


6) In the **Page Designer** pane, **Ctrl+F** to find and replace **localhost:9080** with **<HOSTNAME>:<PORT>** (for example, mvs222.rtp.raleigh.ibm.com:9138).

7) Select **Replace All**.

8) **Ctrl+S** to save the changes and close the Page Designer.

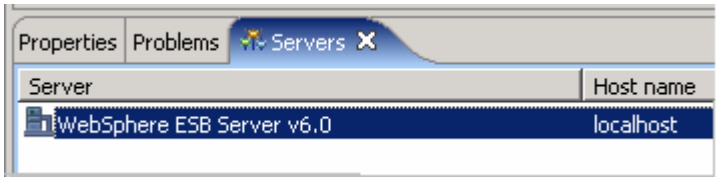
__ j. Switch to the **J2EE perspective** *(Window->Open perspective->Other)* and ensure that the **DynamicEndpointsClient** *Dynamic Web Projects* module and **websphere_default_messaging_provider** *Connecter Projects* are listed as shown below:



Note: Ignore any warnings reflected in the **Problems** view.

__ k. Switch, back to the **Business Integration perspective**

____ 4. Verify that the WebSphere ESB Server V6.0 is listed in the **Servers** view (bottom right window)

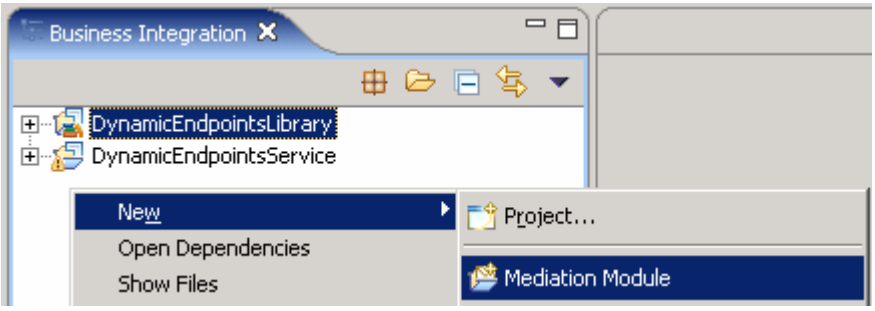


Part 2: Create a mediation module

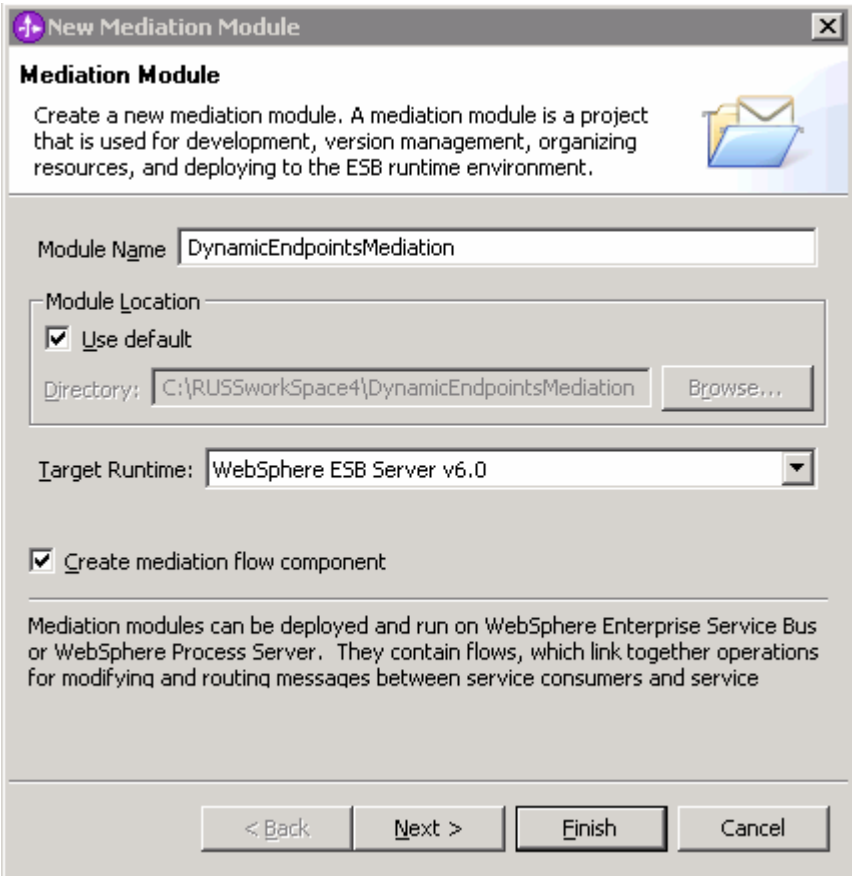
In this section of the lab, a new mediation module is created. There can only be one mediation module for each deployable project.

____ 1. To create the mediation module, complete the following steps:

__ a. In the Business Integration view, right-click to see the context menu and select **New > Mediation Module**. The new Mediation Module window opens

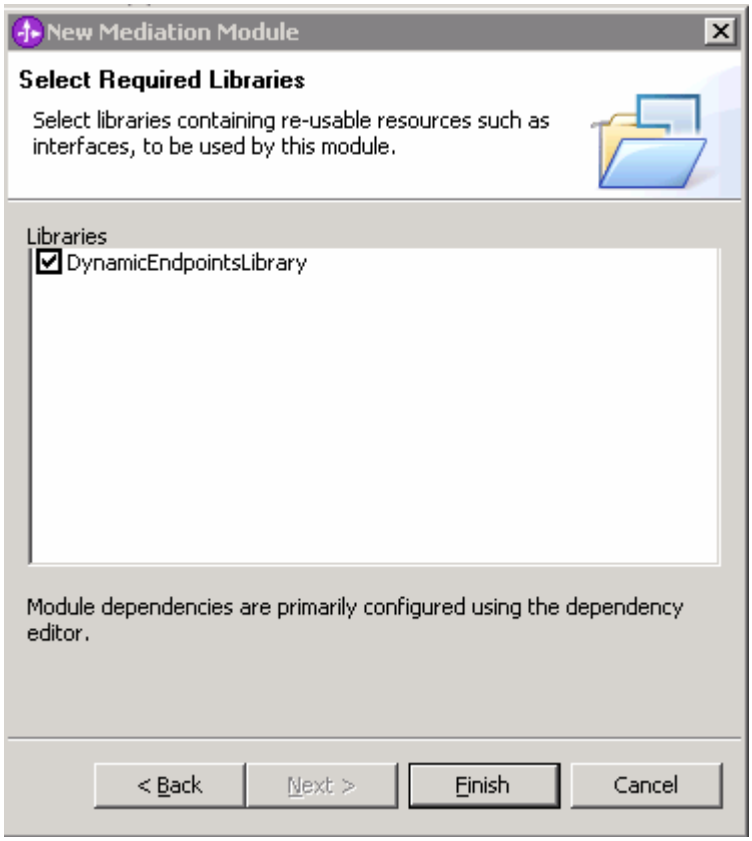


__ b. In the New Mediation Module window, type the **Module Name** as **DynamicEndpointsMediation**



__ c. Verify that the Target Runtime is the WebSphere ESB Server V6.0 and keep the "Create mediation flow component" box checked. Click **Next**

__ d. In the following “Select Required Libraries” wizard, select **DynamicEndpointsLibrary**

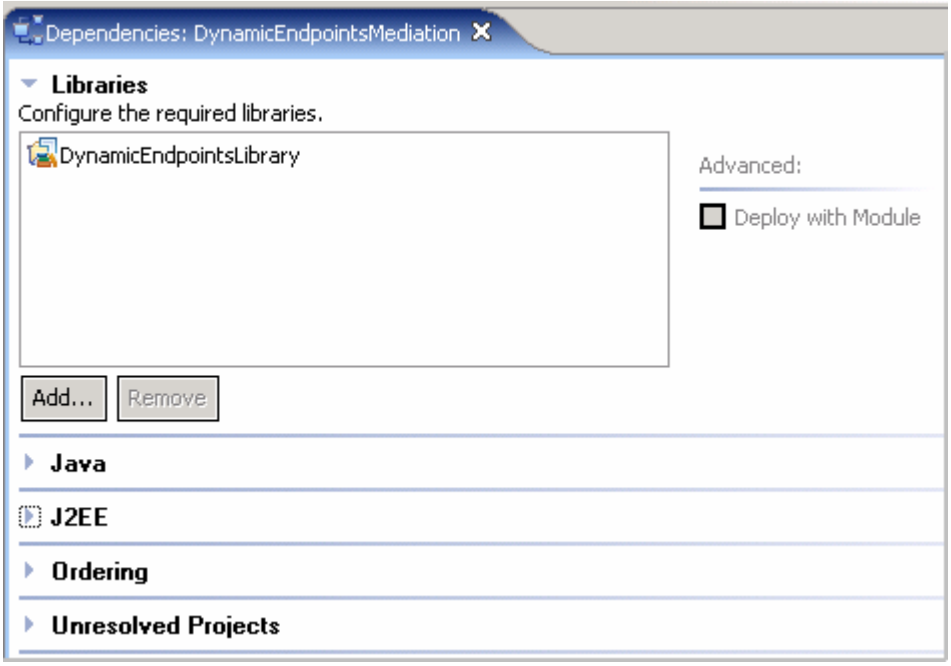


__ e. Click **Finish**

__ f. A mediation module called **DynamicEndpointsMediation** is created. Also a mediation flow component called **Mediation1** is created in the module’s assembly diagram. Ignore any warning messages related to the mediation reflected in the Problems view at this time

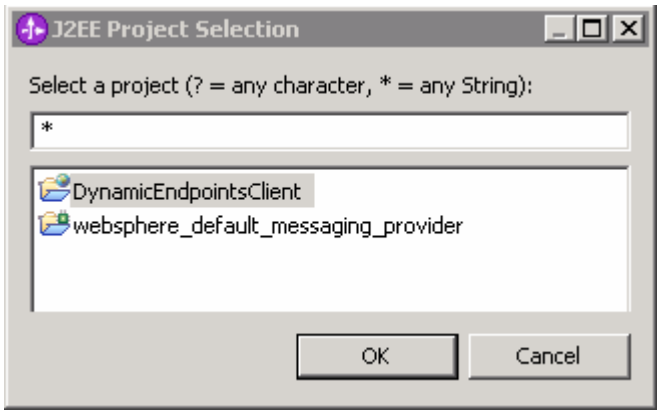
____ 2. Add **DynamicEndpointsClient** Web project and **websphere_default_messaging_provider** connector project to **DynamicEndpointsMediation** as a J2EE dependency

__ a. In the Business Integration view’s tree, expand **DynamicEndpointsMediation** module (if not already expanded) and double-click on Dependencies (Dependencies) to open the Dependency Editor



__ b. In the Dependency Editor, expand **J2EE** and click the **Add** button to add **DynamicEndpointsClient** and **websphere_default_messaging_provider** as J2EE dependent projects for the mediation module

__ c. From the **J2EE Project Selection** window, select the projects listed one at a time and click **OK**

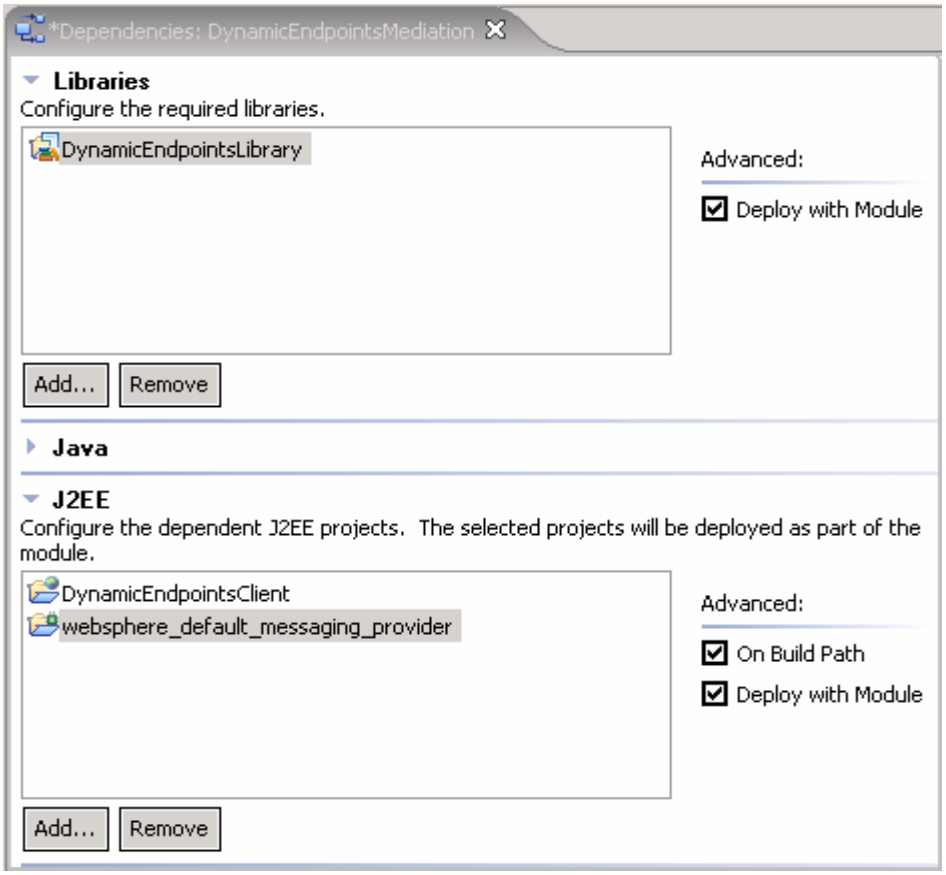


__ d. Ensure that the check box next to **Deploy with Module** is selected by selecting each of the modules added

---

NOTE: The required configured **DynamicEndpointsLibrary** was added as a dependent library while creating the mediation module

---

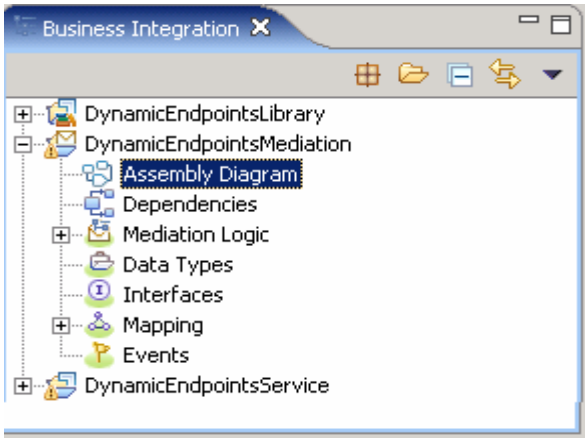


__ e. **Ctrl+S** to save the changes and close the Dependency Editor

____ 3. Open the **DynamicEndpointsMediation** module assembly with the Assembly Editor to visually compose the Mediation Module

__ a. Open **DynamicEndpointsMediation** module assembly

1) In the Business Integration view's tree, expand the **DynamicEndpointsMediation** module (if not already expanded) and double-click on Assembly Diagram (Assembly Diagram) to open it with the assembly editor



2) Notice that a default mediation flow was created and is named Mediation1

__ b. Add a **Stand-alone Reference** to the Mediation in the Assembly Diagram

1) Add a **Stand-alone Reference** from the palette( ) and drop it on the left-hand side of **Mediation1**

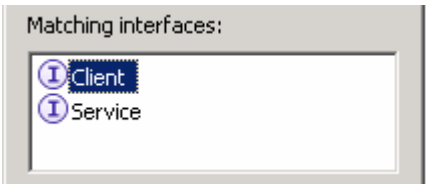
a) Select the **Stand-alone Reference** icon from Assembly Diagram's palette tray ( )

(**Note: The Assembly Editor doesn't show all the menu options. Click on the up and down arrows on the palette tray to see the rest of the menu options)**

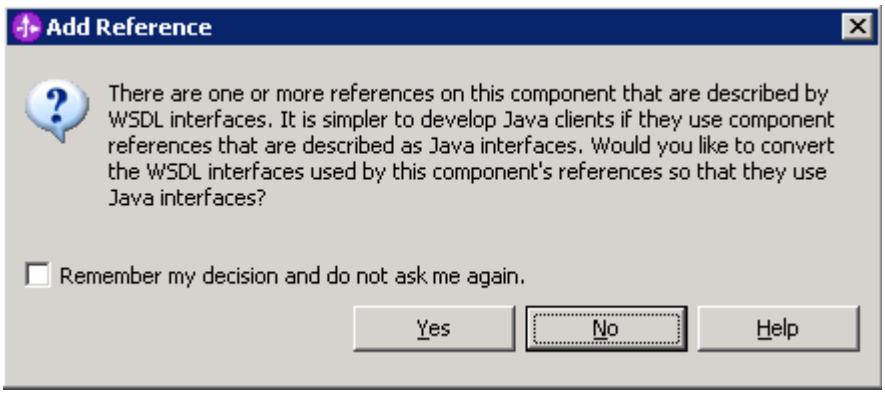
b) Click or drag the **Stand-alone Reference** and drop it to the left side of **Mediation1**

2) Hover over the Stand-alone Reference or click on the Stand-alone Reference so the Add Reference icon appears. Click the icon ( ) to add a reference

3) From the **Add Reference** window, select **Client** (if not already selected) from the list of matching interfaces and accept the default Name



4) Click **OK**



5) Click **NO** to the Question on the pop-up dialog

6) The **Stand-alone Reference** must look as below:

__ c. Add an Import to the Mediation in the Assembly Diagram

1) Add an **Import** from the palette( ) and drop it on the right-hand side of **Mediation1**

a) Click on **Import** icon from Assembly Diagram tray ( )

b) Click or drag then Import icon and drop it to right side of **Mediation1**

2) Right click over the import and select **Rename** from the context menu to change default import name from Import1 to **WiredImport**

3) Hover over or click on WiredImport so the add interface icon appears. Click the icon ( ) to add an interface

4) Select **Service** from the list of matching interfaces

__ d. Click **OK**

__ e. Add another Import to Mediation in the Assembly Diagram

1) Add an **Import** from the palette( ) and drop it on the right-hand side of **Mediation1**

a) Click on **Import** icon from Assembly Diagram tray ( )

b) Click or drag the Import icon and drop it to the right side of **Mediation1**

2) Right click over the import and select **Rename** from the context menu to change the default import name from Import1 to **JmsImport**

3) Hover over or click on the JmsImport so the add interface icon appears. Click the icon ( ) to add an interface

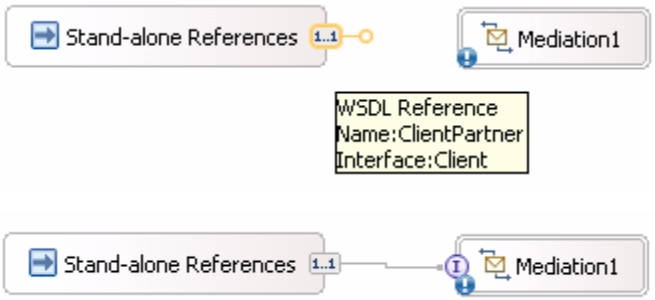
4) Select **Service** from the list of matching interfaces

5) Click **OK**

6) The Assembly Diagram looks like the one shown below:

__ f. Wire components together

1) Click on **Stand-alone Reference** and ensure the **ClientPartner Reference** node is selected and drag a wire and connect to **Mediation1**. Click **OK** to any pop-up dialog



2) Click on **Mediation1** and drag a wire and connect to **WiredImport**. Click **OK** to any pop-up dialog

3) Leave the **JmsImport** component alone

__ g. Generate SCA bindings for the Imports

1) Right-click on **WiredImport** and select **Generate Binding → SCA Binding** from the context menu

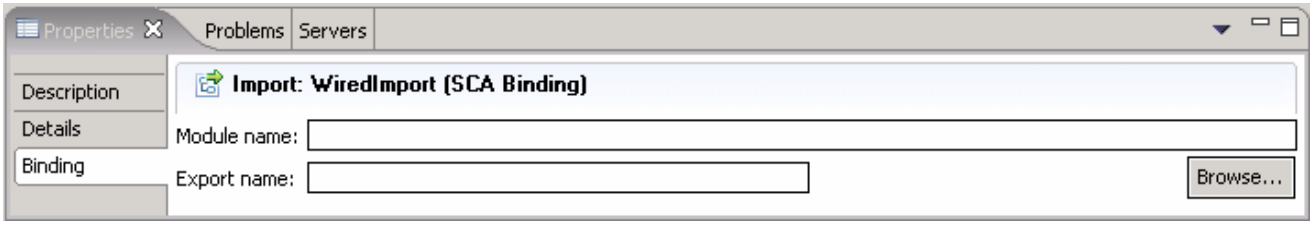
2) Right-click on **JmsImport** and select **Generate Binding → Message Binding → JMS Binding** from the context menu. In the **JMS import binding** window, accept the default JMS Binding configuration properties and click **OK**

3) This is how the Assembly Diagram looks after generating the bindings:

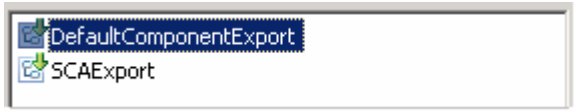
__ h. Update Binding for the **WiredImport**

1) Select the **WiredImport** component in the Assembly Diagram

2) In the Properties view (bottom right window) select the **Binding** tab

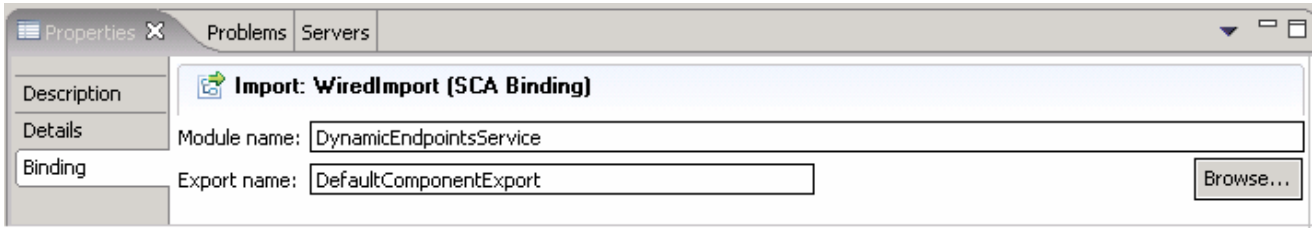


3) Click the browse button next to **Export name** field and select **DefaultComponentExport**



4) Click **OK**

5) The **Module name** and **Export name** must be populated as **DynamicEndpointsService** and **DefaultComponentExport** respectively



6) Save all work by choosing **File > Save All** or **Crtl + Shift + S**

__ i. Configure End-point Configuration binding for the **JmsImport**

1) Select the **JmsImport** component in the Assembly Diagram

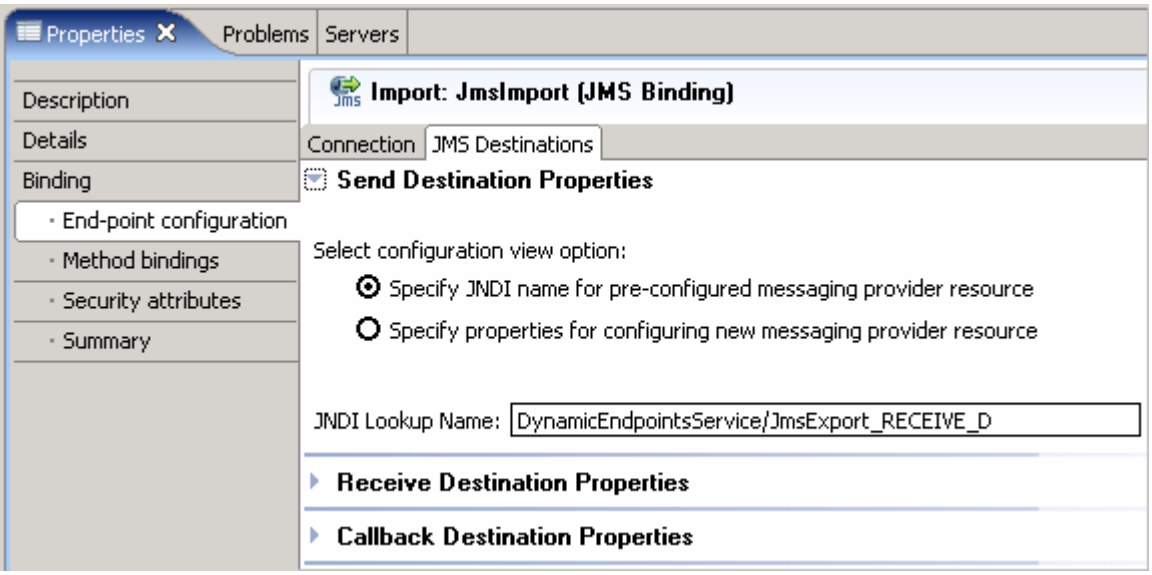
2) In the Properties view select **Binding tab → End-point configuration**

3) Select the **JMS Destinations** tab

4) Expand **Send Destination Properties**

5) Select the **Specify JNDI Name for pre-configured messaging provider resource** for the configuration view option

6) Enter **DynamicEndpointsService/JmsExport_RECEIVE_D** for JNDI Lookup name



7) Save all work by choosing **File > Save All** or **Crtl + Shift + S**

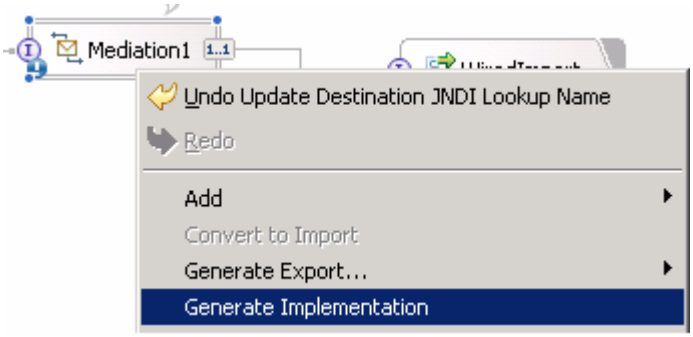
Part 3: Generate mediation flow implementation for the mediation

In this section of the lab, the Mediation component is selected for generating the implementation and such an action opens the implemented Module in a Mediation Flow Editor. On connecting the source and target operations for the Operations Connections view, the mediation primitive, that is, an XSL Transformation primitive is added to the Request and Response flows.

____ 1. To open the Mediation Flow Editor for the mediation module, **Mediation1,** follow the steps below:

__ a. Open the mediation module in the assembly editor

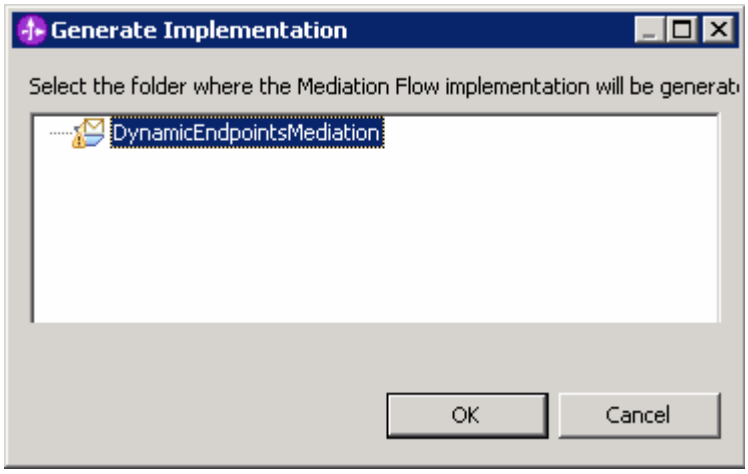
__ b. In the Mediation Module's Assemble diagram, right click on **Mediation1** and choose **Generate Implementation** from the context menu



Alternative 1: Double-click on the mediation flow component, **Mediation1**. If an implementation exists, it will be opened in the mediation flow editor. If the component is not implemented, a dialog will pop up asking to confirm for implementation. Click **Yes**, and select the target mediation module, **DynamicEndpointsMediation.** Click **Finish**

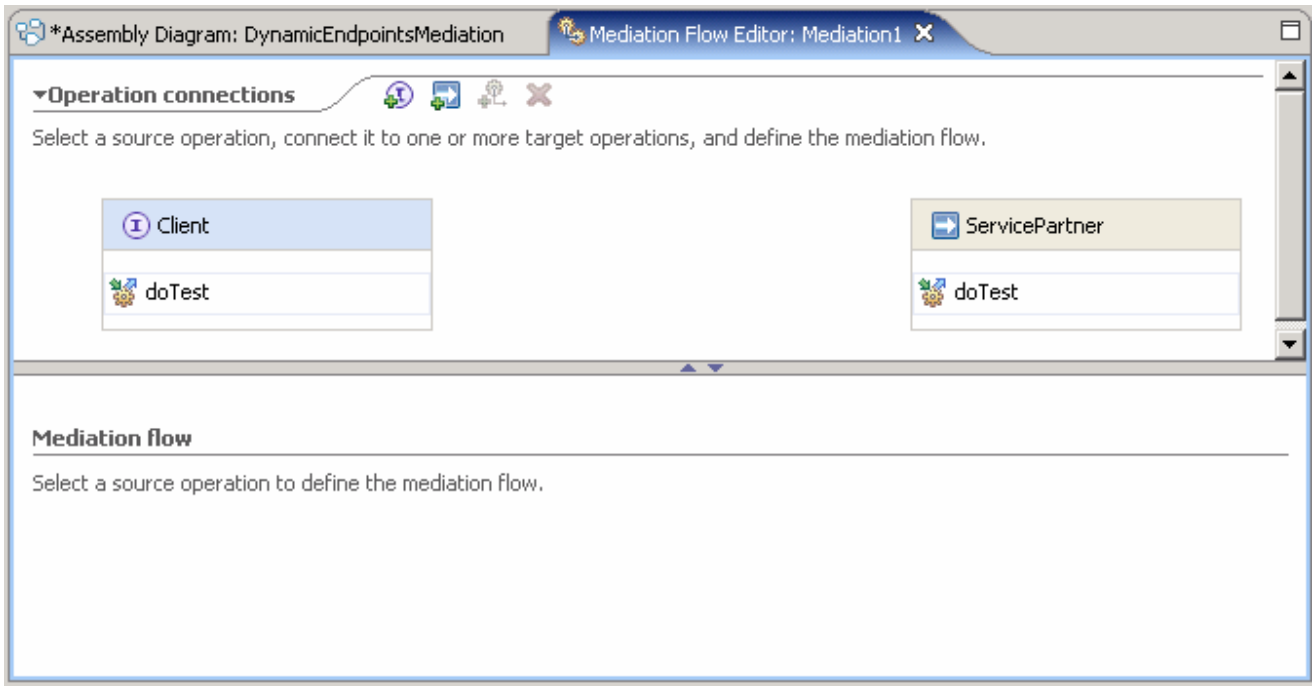
Alternative 2*:* In the Business Integration View, expand the mediation module, **DynamicEndpointsMediation** and then expand the **Flows** category. Select the mediation flow, **Mediation1**, right-click and select **Open**

__ c. Select the target mediation module, **DynamicEndpointsMediation** from the Generate Implementation dialog



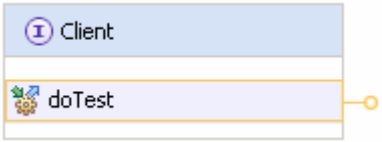
__ d. Click **OK**

__ e. The **Mediation Flow Editor** will open after generating implementation for the mediation module, **Mediation1**



____ 2. Connect the **source** to **target operations** in Operation Connections view

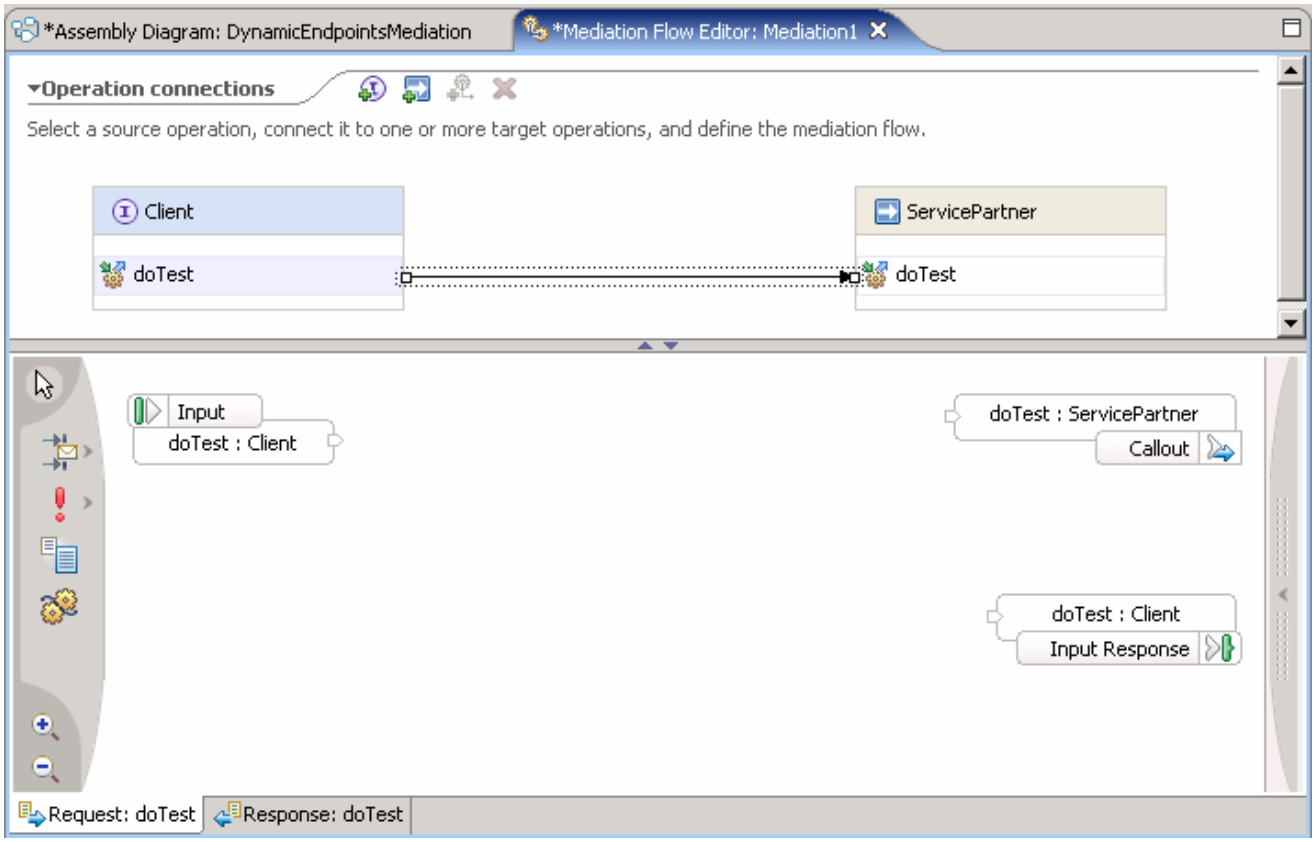
__ a. Click anywhere on the source operation, **Client/doTest** on the left-hand side of the Operation Connections view



__ b. Drag to the target operation, **ServicePartner/doTest** on the right-hand side of the Operation Connections view and release the mouse click

Alternative: Right-click on the source component, **Client/doTest** and select **Create an operation connection** and then click on the target operation, **ServicePartner/doTest**

____ 3. Click on the black line (wire) to view the Mediation Flow View and ensure the **Request** tab is selected to build the Request flow as shown below

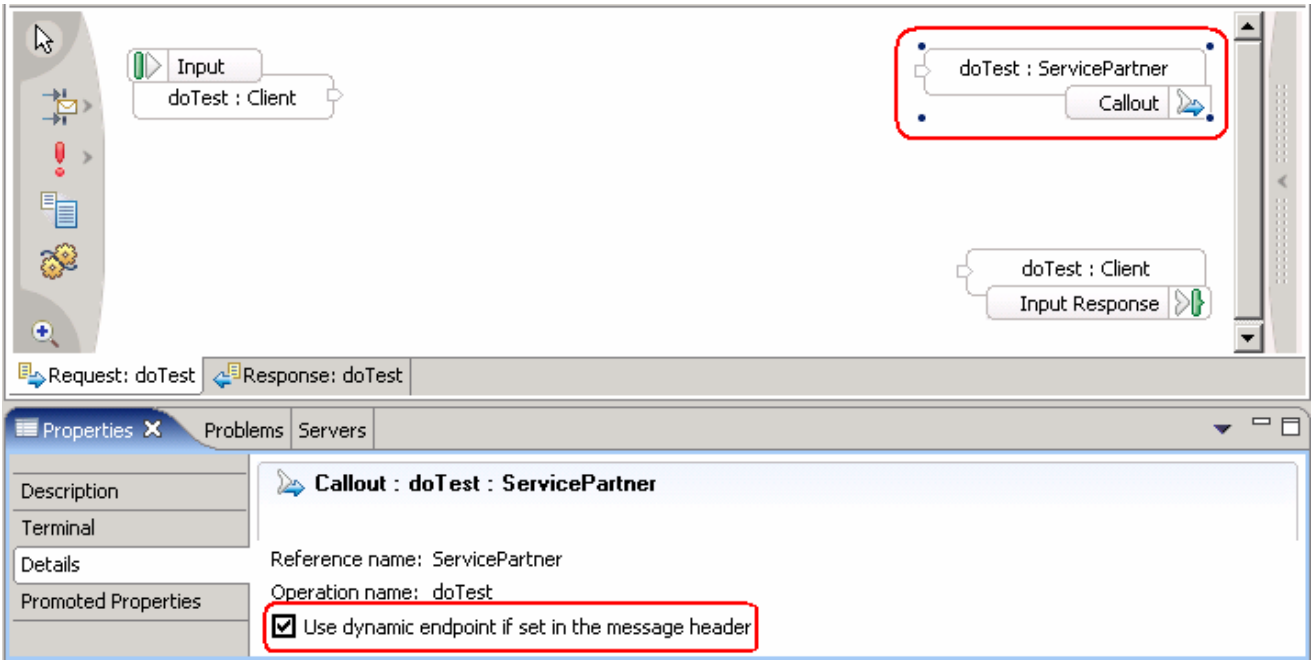


____ 4. Ensure that **Callout** property is enabled

__ a. In the Mediation Flow Editor (Request flow - middle window (doTest:ServicePartner)), select the **Callout node** and select the **Details** tab under its Properties view (bottom window) to ensure the "**Use dynamic endpoint if set in the message header**" is selected

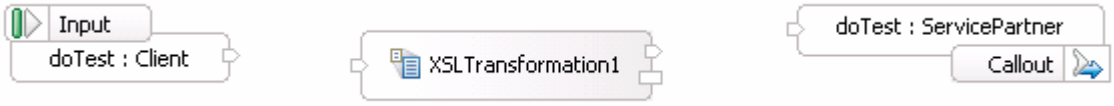
---

NOTE: By default the "**Use dynamic endpoint if set in the message header"** property is enabled.

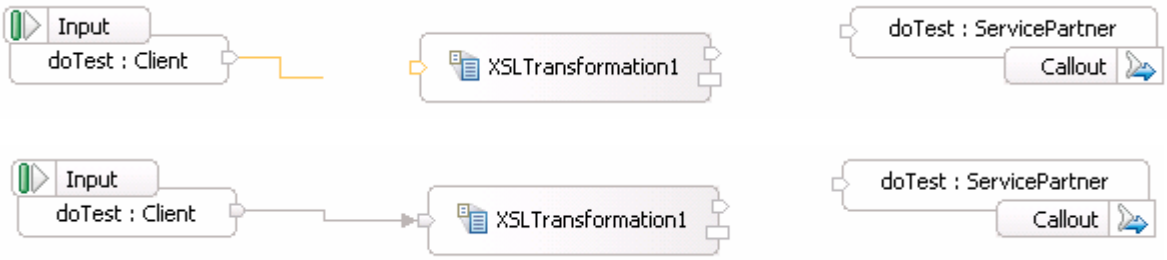


____ 5. Add a **XSL Transformation** primitive to the Request Mediation Flow diagram

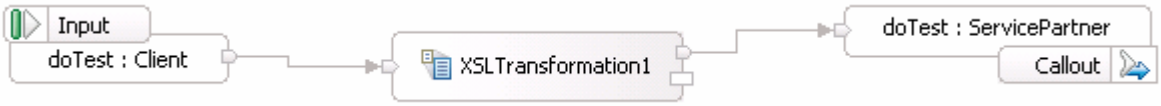
__ a. In the **Mediation Flow Editor**(middle), click on **XSL Transformation** icon ( ) to select the XSL Transformation primitive from the pallet on left-hand side and drop it into the canvas between the Input Request node and the Callout Request Node



__ b. Hover the mouse over **Input node**'s output terminal and drag the handle that appears to the input terminal of the XSL Transformation primitive, **XSL Transformation1**

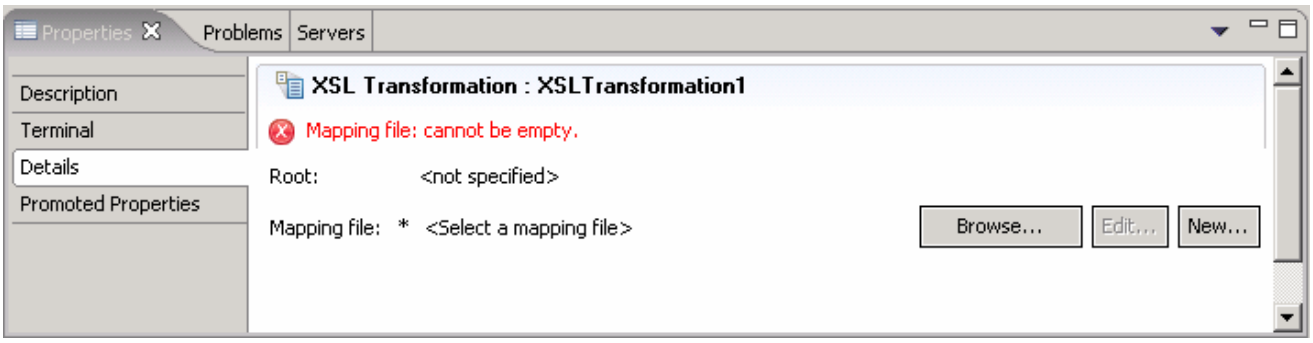


__ c. Hover the mouse over XSL Transformation primitive, **XSL Transformation1**'s output terminal and drag the handle that appears to the input terminal of the **Callout node**



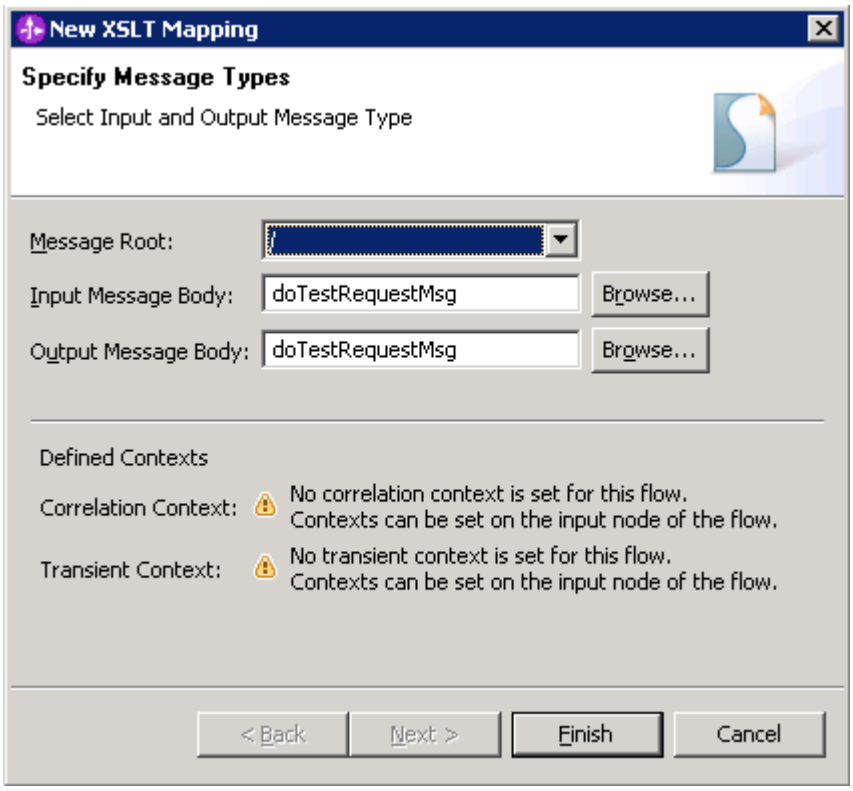
____ 6. Set the Properties for the XSL Transformation primitive, **XSL Transformation1**

__ a. In the Mediation flow editor, select the XSL Transformation primitive, **XSL Transformation1** and choose the **Details** tab under its properties view



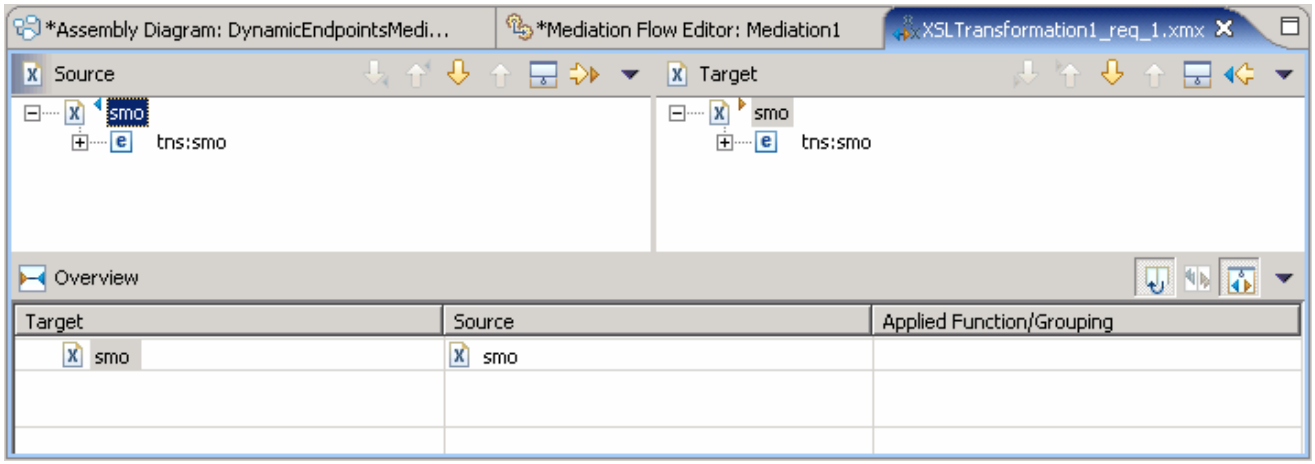
__ b. In the **Mapping file** filed, click **New** to create a new XSL map using the mapping editor

__ c. In the **New XSL Mapping** dialog, select the **Message Type** as ***/*** (Root) to transform the complete message, from the drop down list. Accept the **Input Message Body** and **Output Message Body** as defaults

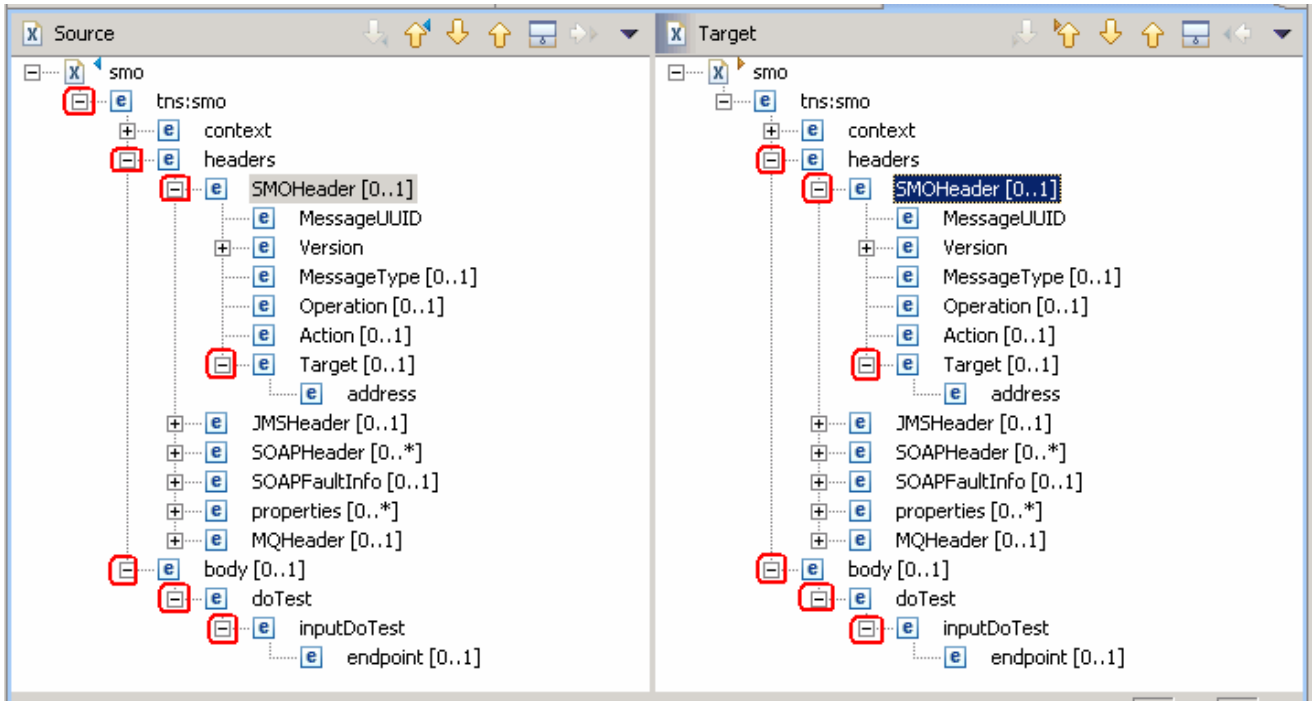


__ d. Click **Finish**

__ e. The XSL Transformation mapping editor opens

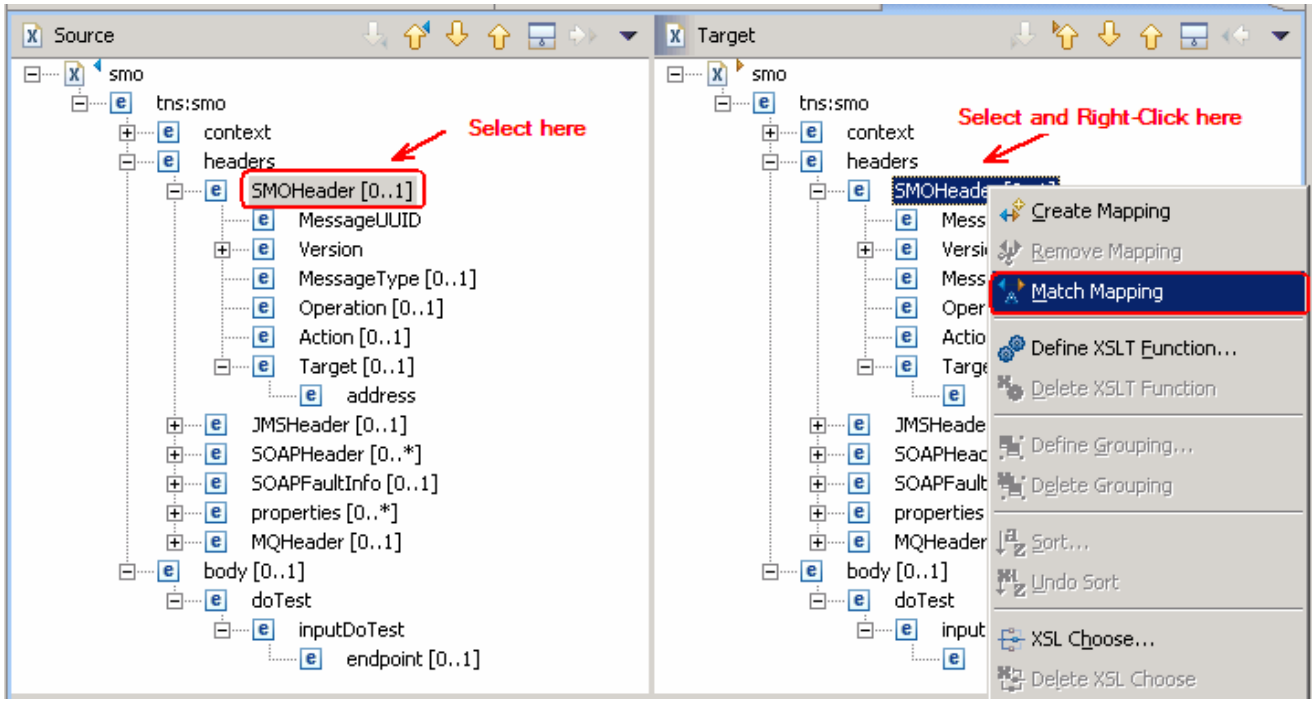


__ f. Expand the Header and Body message trees for the Source and the Header and Body message trees for the Target of the mapping editor, exactly as shown below:

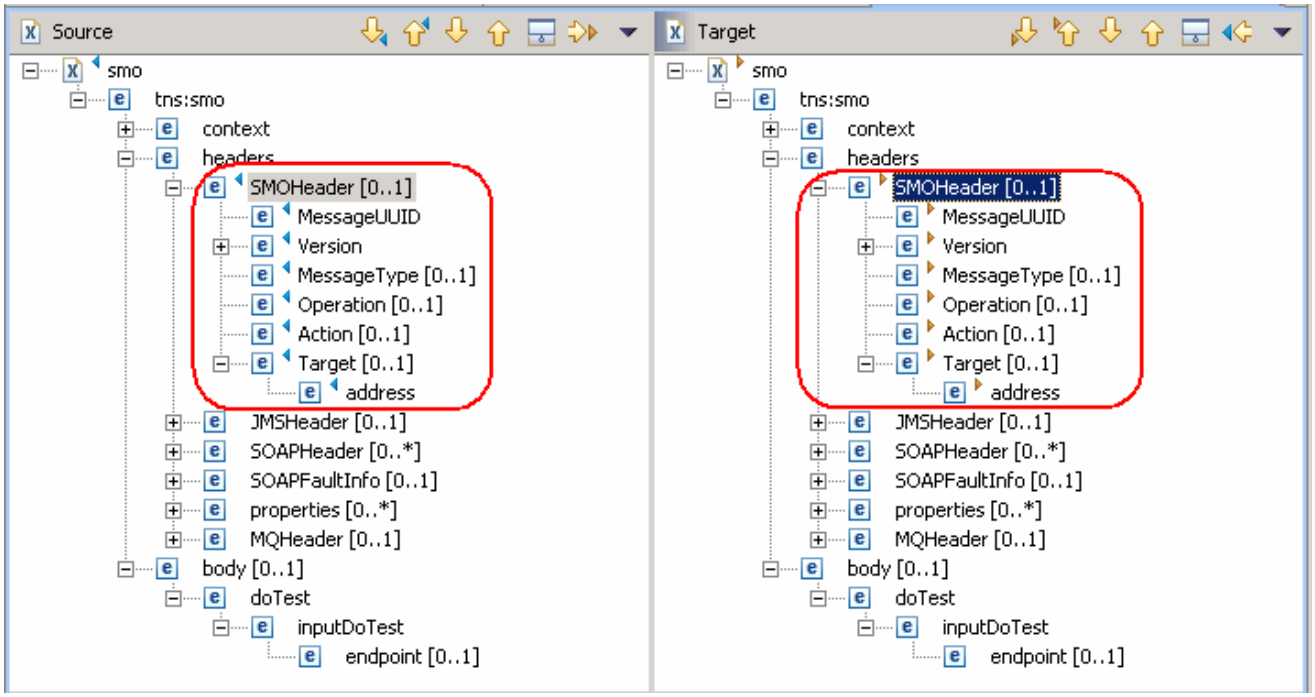


__ g. Map the following sources and targets:

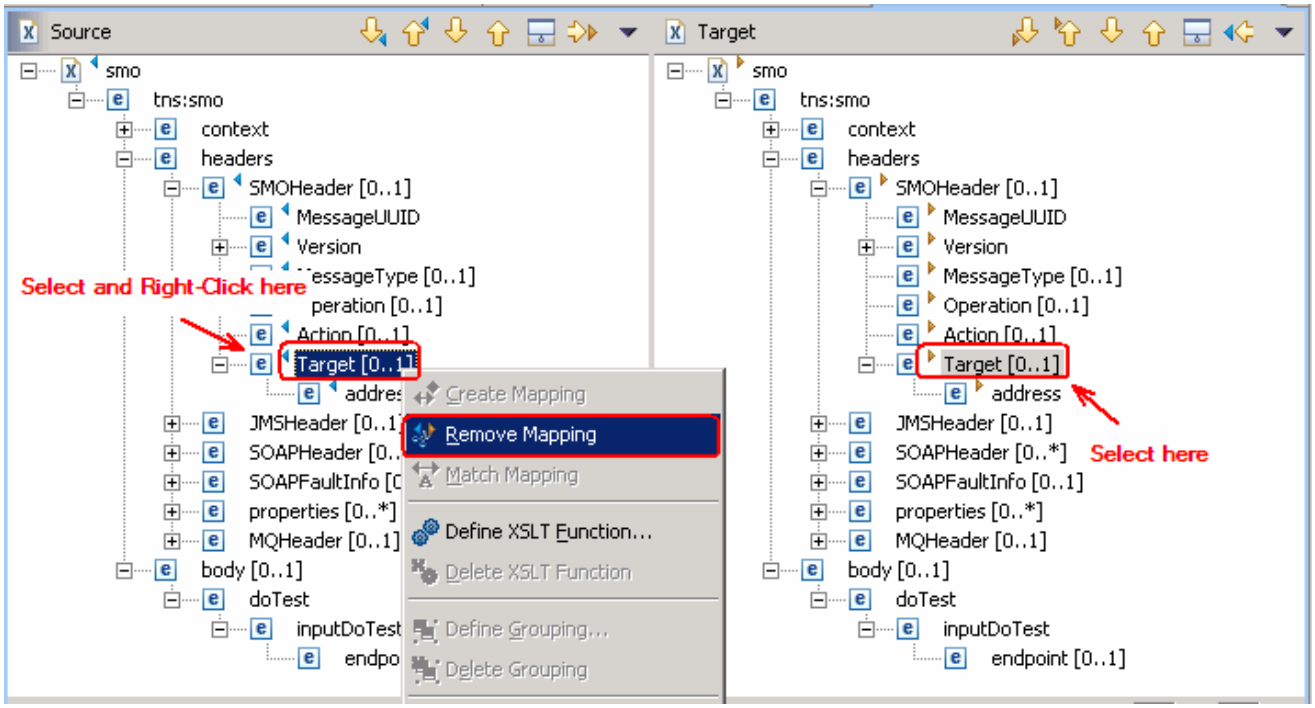
1) Match mapping the **SMOHeader** from source to target. Select the Header element, **SMOHeader** in the **Source** section and then select and right-click on the Header element **SMOHeader** in the target section and select **Match Mapping** from the context menu as shown below:



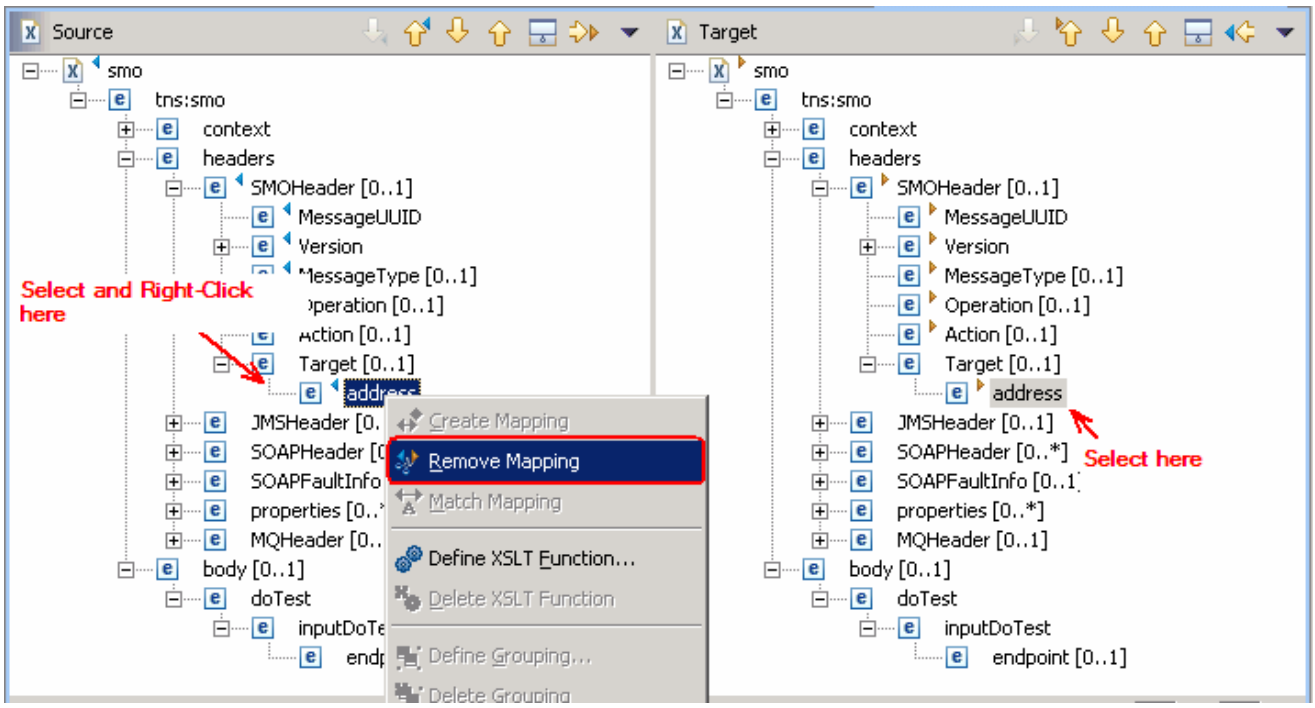
2) The result is a as shown below:



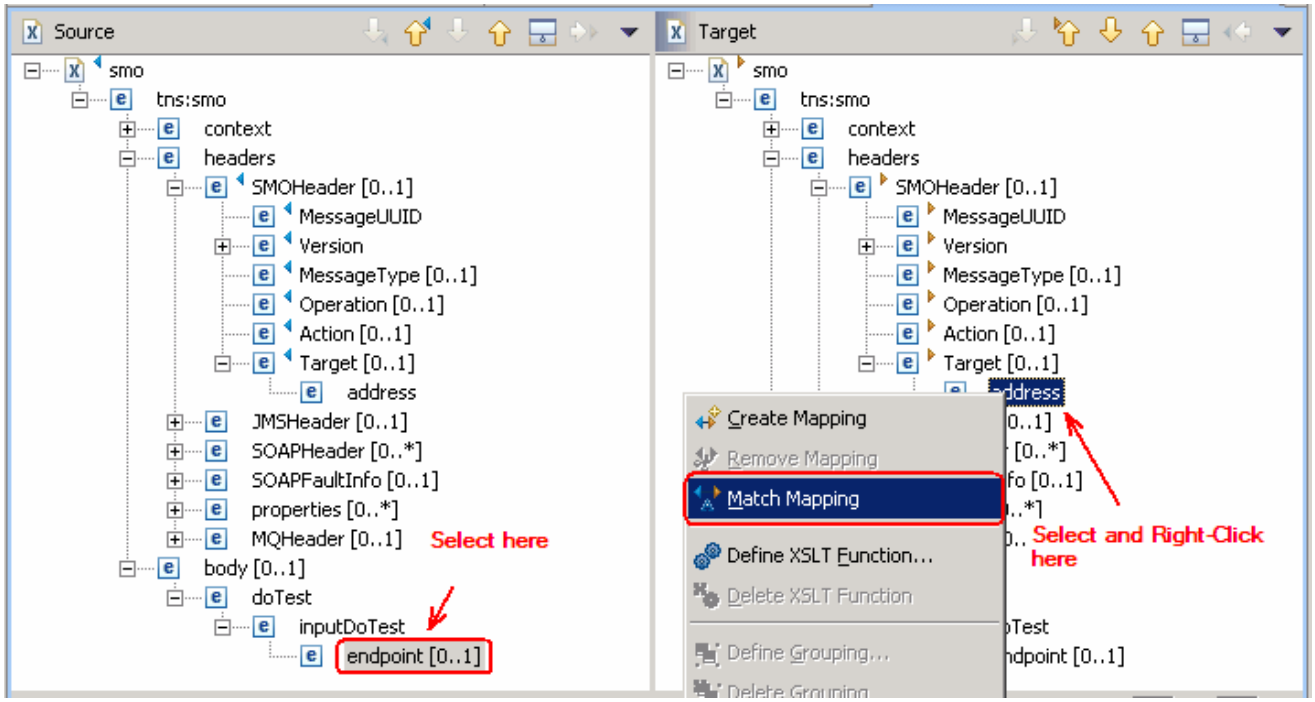
3) Now **Remove Mapping** for the **Target[0..1]** element as shown below:



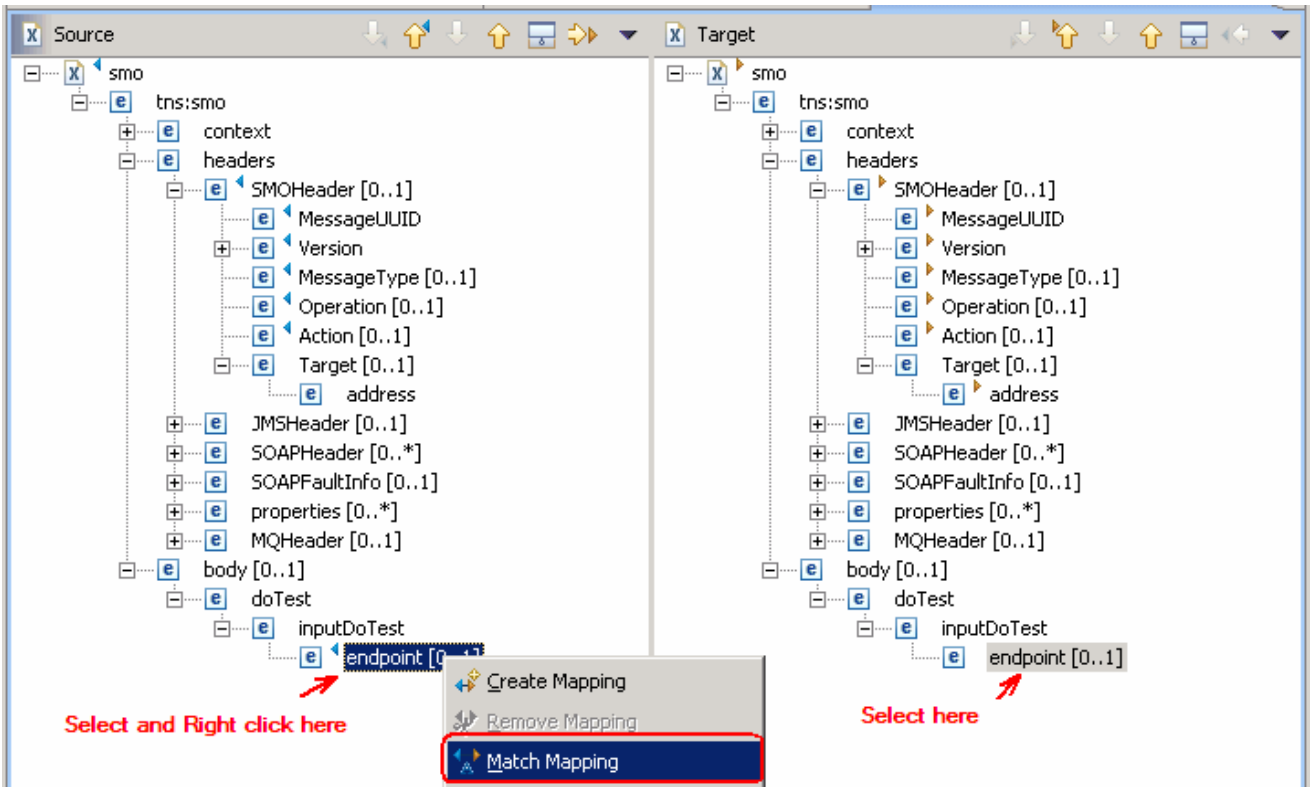
4) Now **Remove Mapping** for the **address** element as shown below:



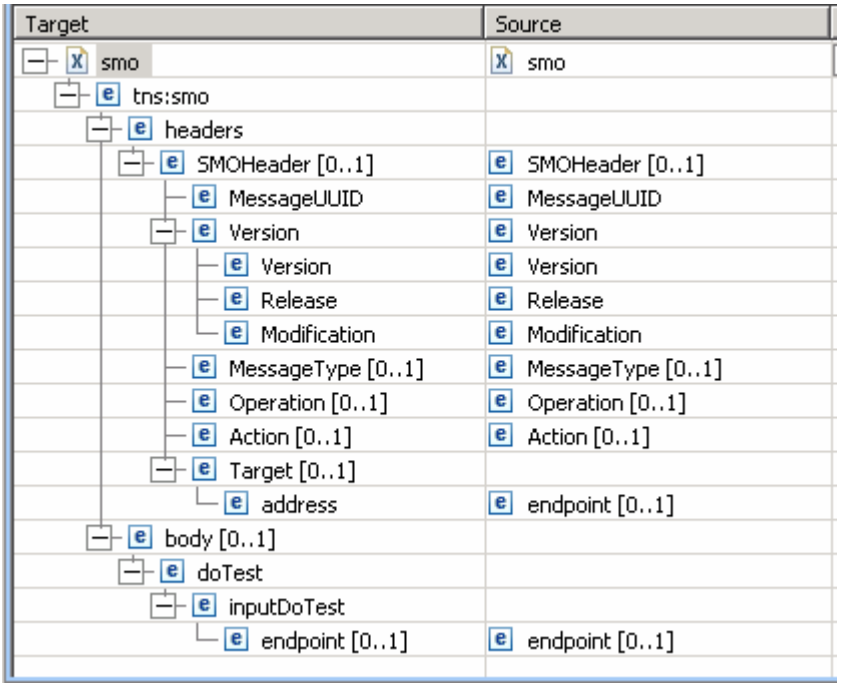
5) Now, Match Mapping for the **endpoint** body element in the Source section to the **address** element in the Target section as shown below:



6) Map Match the **endpoint** element source body to the target body **endpoint** element



__ h. The final **Overview** (middle window) of the mapped elements must look exactly like diagram shown below:

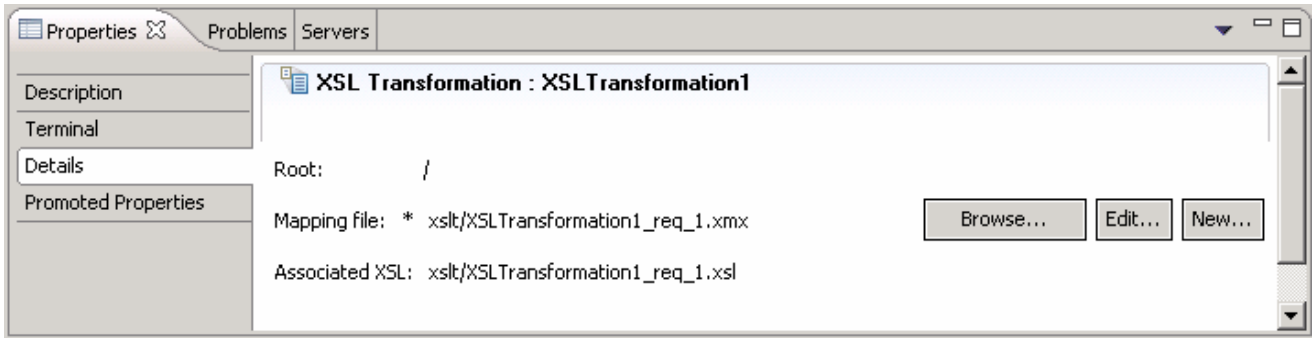


Note: The automatic generation capability of the associated XSL style sheet is enabled; the XSL and XMX files will automatically stay synchronized.

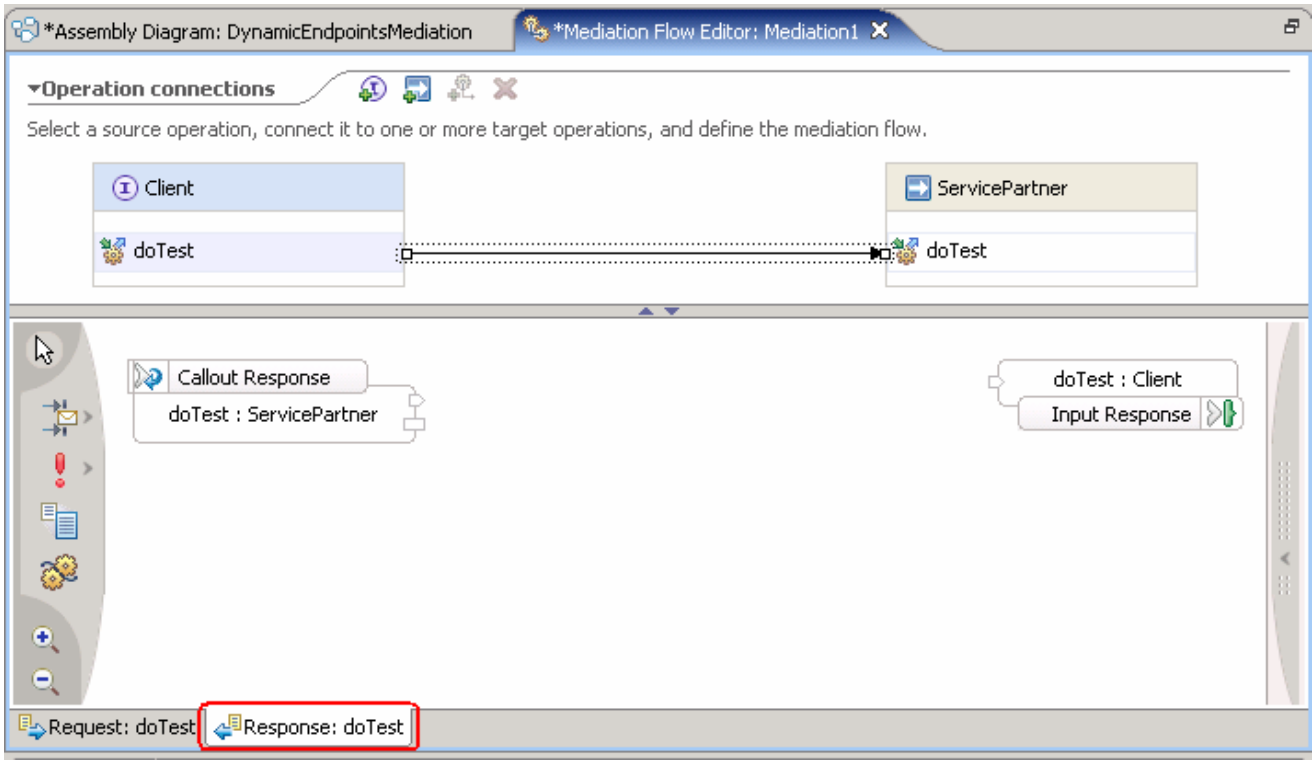
__ i. Save all work by choosing **File > Save All** or **Crtl + Shift + S**

__ j. Close the XSL transformation window.

__ k. On the Mediation Flow Editor, the XSL Transformation Binding must look as shown below:

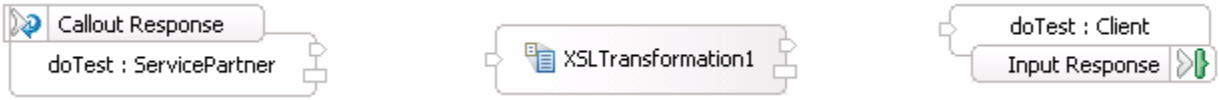


____ 7. Click on the black line (wire) to view the Mediation Flow View and select the **Response** tab to build the Response flow as shown below:

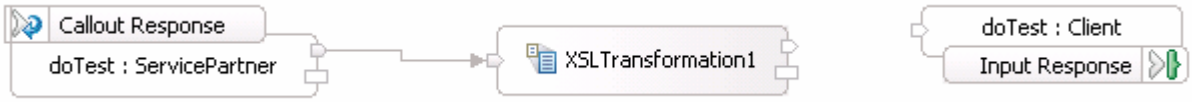


____ 8. Add **XSL Transformation** primitive to Response Mediation Flow diagram

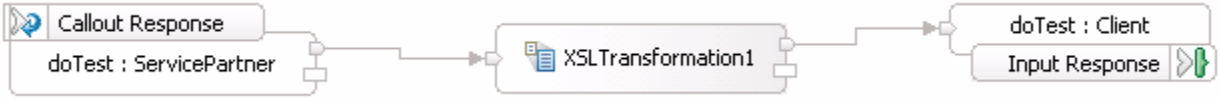
__ a. In the **Mediation Flow Editor**(middle), click on **XSL Transformation** icon ( ) to select the XSL Transformation primitive from the pallet on left-hand side of view and drop it into the canvas between the Callout Response node and the Input Response Node



__ b. Hover the mouse over **Callout Response** node's output terminal and drag the handle that appears to the input terminal of the XSL Transformation primitive, **XSL Transformation1**

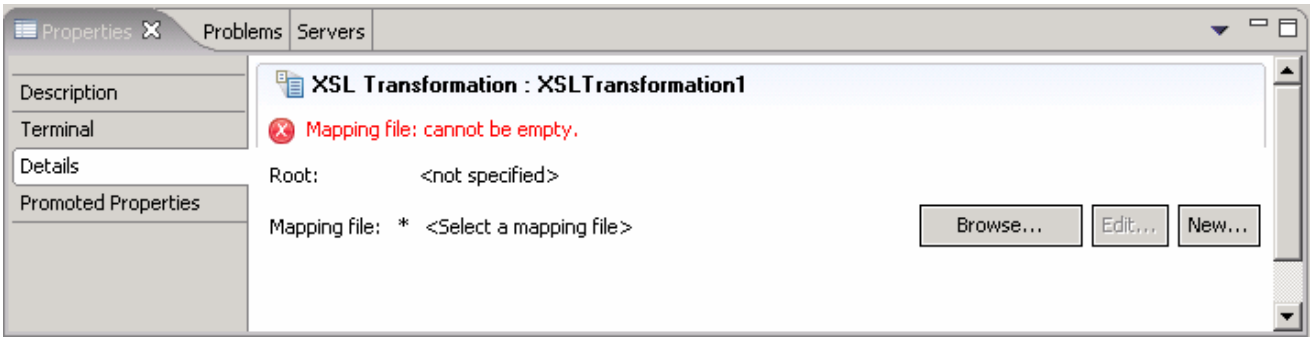


__ c. Hover the mouse over XSL Transformation primitive, **XSL Transformation1**'s output terminal and drag the handle that appears to the input terminal of the **Input Response node**



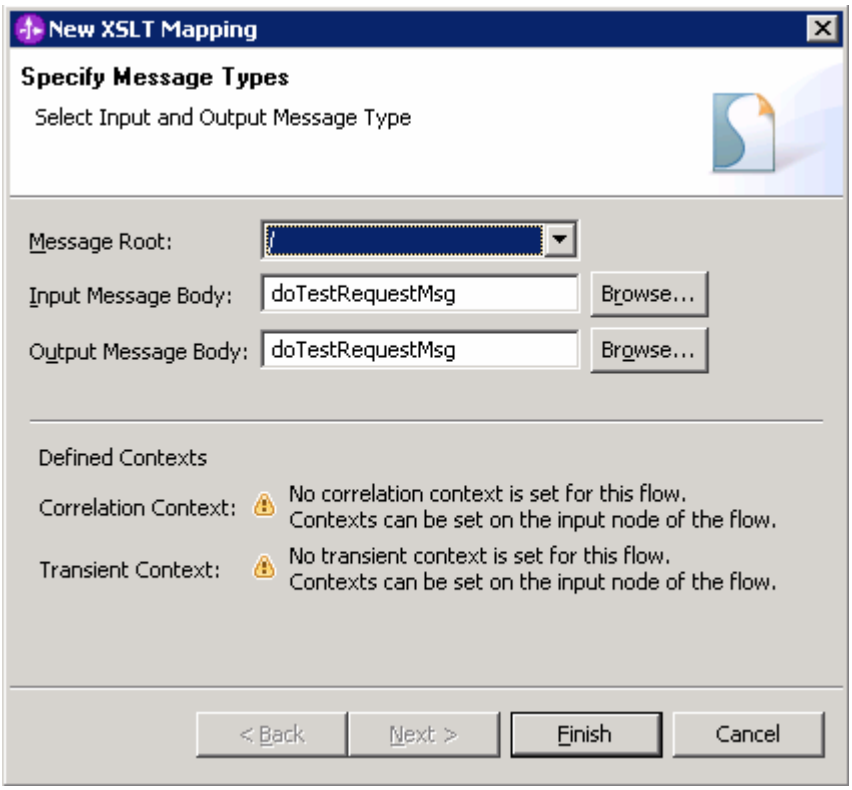
____ 9. Set the Properties for the XSL Transformation primitive, **XSL Transformation1**

__ a. In the Response Mediation flow editor, select the XSL Transformation primitive, **XSL Transformation1** and choose **Details** under its properties view



__ b. In the **Mapping file** filed, click **New** to create a new XSL map using the mapping editor

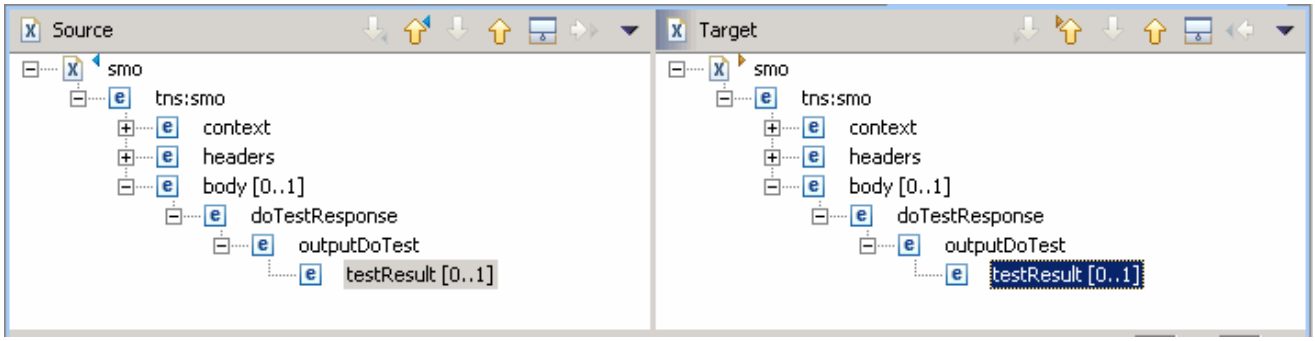
__ c. In the **New XSL Mapping** dialog, select the **Message Type** as ***/*** (Root) to transform the complete message, from the drop down list. Accept the **Input Message Body** and **Output Message Body** as defaults



__ d. Click **Finish**

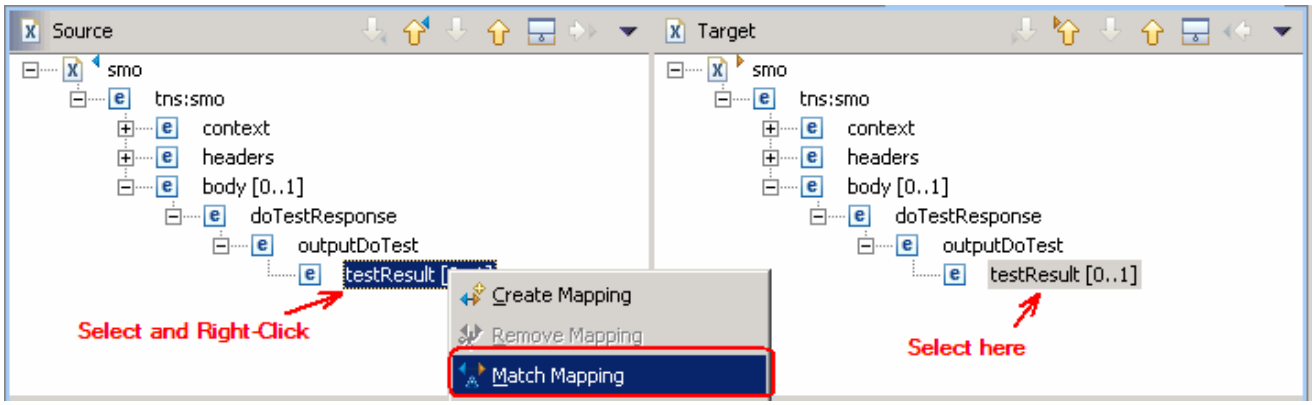
__ e. The XSL Transformation mapping editor opens

__ f. Expand only the Body message trees in the source and target of the mapping editor as shown below:



Map the body **testResult** element from source to target

__ g. Map Match the **testResult** element of the source body and **testResult** element of the target body

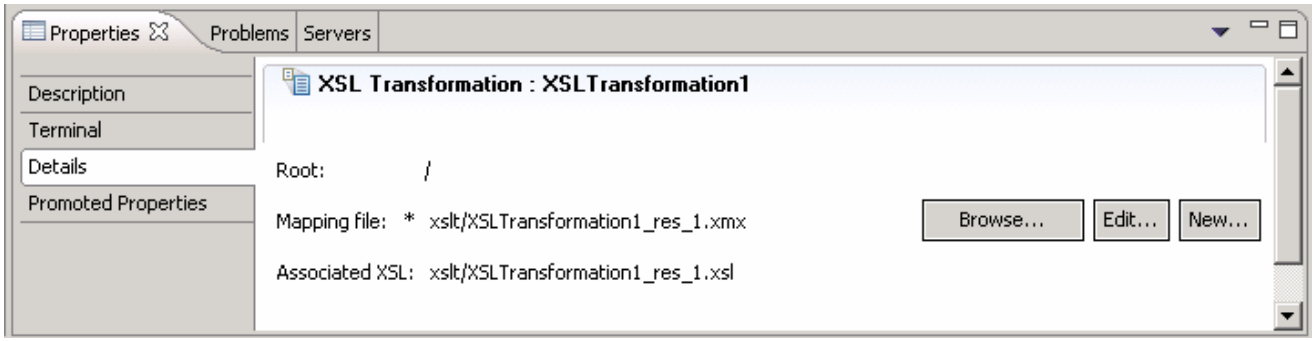


__ h. Save all work by choosing **File > Save All** or **Crtl + Shift + S**

__ i. Close XSL transformation Editor

Note: The automatic generation capability of the associated XSL style sheet is enabled, the XSL and XMX files will automatically stay synchronized.

__ j. In the Mediation Flow Editor, the XSL Transformation Binding must look as shown below:



____ 10. **Close** the Mediation Flow Editor

Part 4: Test dynamic end points

In this section of the lab, a JSP client is used test various service endpoints with the Dynamic End Point property enabled and disabled for the Callout

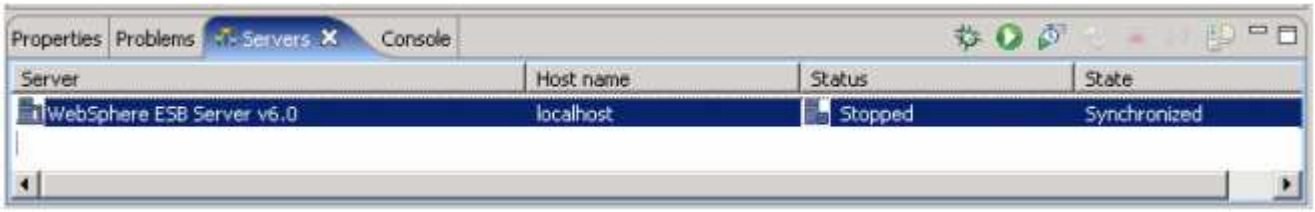
____ 1. Start **WebSphere ESB Server** and **add modules** to server

If using a remote testing environment, follow the instructions in **Task: Adding remote server to WebSphere Integration Developer test environment** at the end of this document, to start the remote server.

If using a local testing environment:

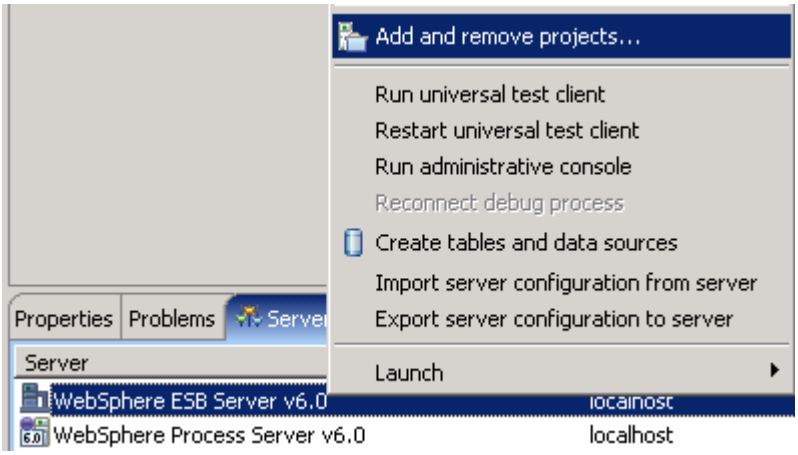
__ a. Open **Servers View** (bottom right window)

__ b. Select the WebSphere ESB Server V6.0 and right-click to select "( ) **Start**" from the context menu



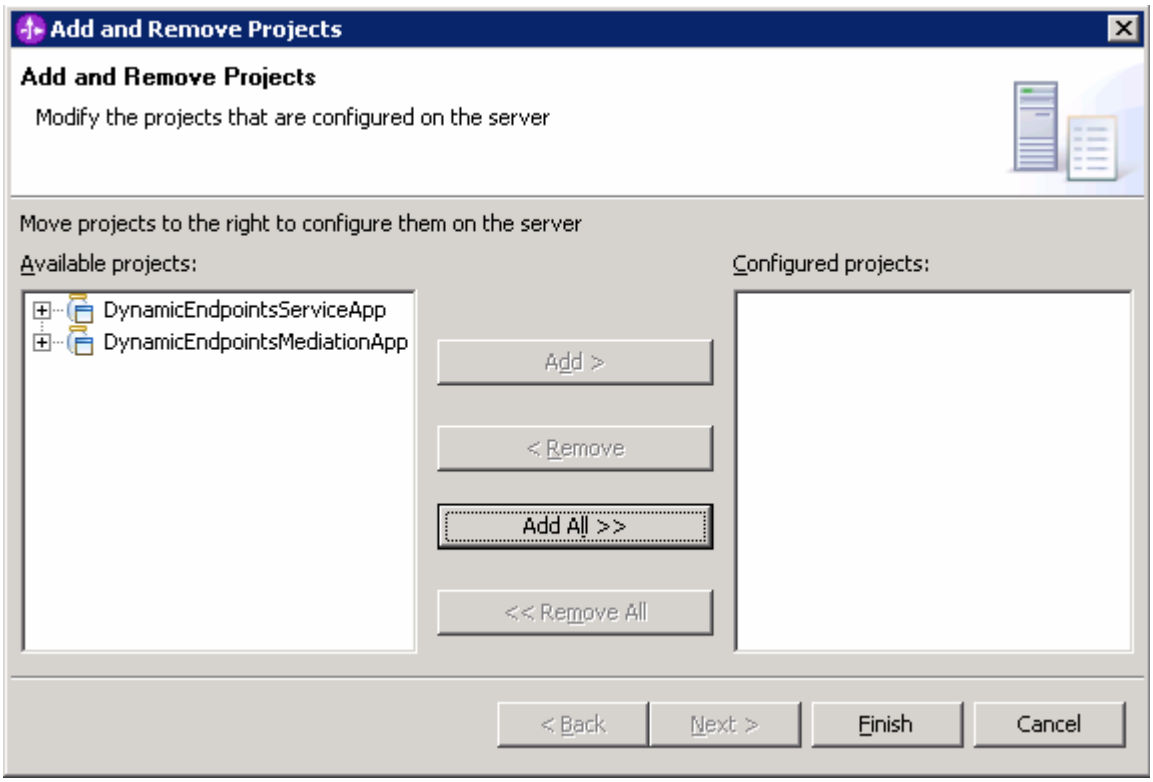
__ c. This takes some time. Wait for the server to start

__ d. Add projects to WebSphere ESB Server. In Servers view, right-click on WebSphere ESB Server V6.0 and select "**Add and Remove Projects...**" from the context menu



***NOTE*:** Note that the ESB server that is being used is configured with an ESB profile that is part of the installation and not part of the workspace. Therefore, if there are any projects deployed to the server from a different workspace, there may be some naming conflicts or other problems. If this occurs, open the Administrative Console and stop/uninstall those projects before adding these projects. This should avoid any potential errors.

__ e. The available projects are listed as shown below:



__ f. Click **Add-All>>** button to move all projects to server

__ g. Click **Finish**

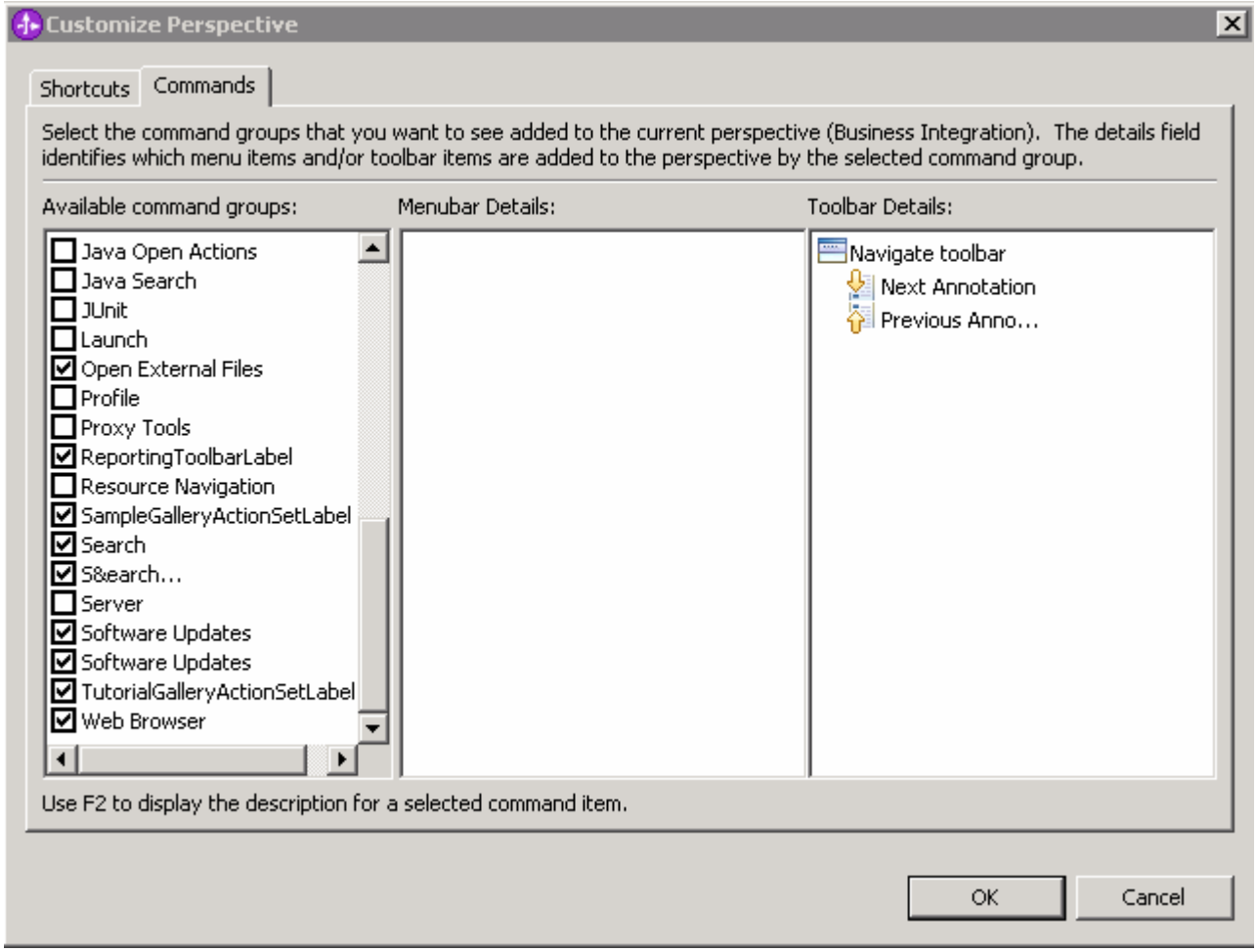
__ h. Wait for the deployment to finish. While the project is deploying you will see something like the following in the lower right corner of WebSphere Integration Developer

____ 3. Enable the Web browser in the WebSphere Integration Developer

__ a. In the main menu select **Window > Customize Perspective**

__ b. Select the **Commands** tab and scroll to the bottom

__ c. Select the check box next to **Web browser**



__ d. Click **OK**

__ e. This should put an icon for Web browser in the WebSphere Integration Developer tools panel

__ f. Click on the Web browser icon to launch a browser in WebSphere Integration Developer

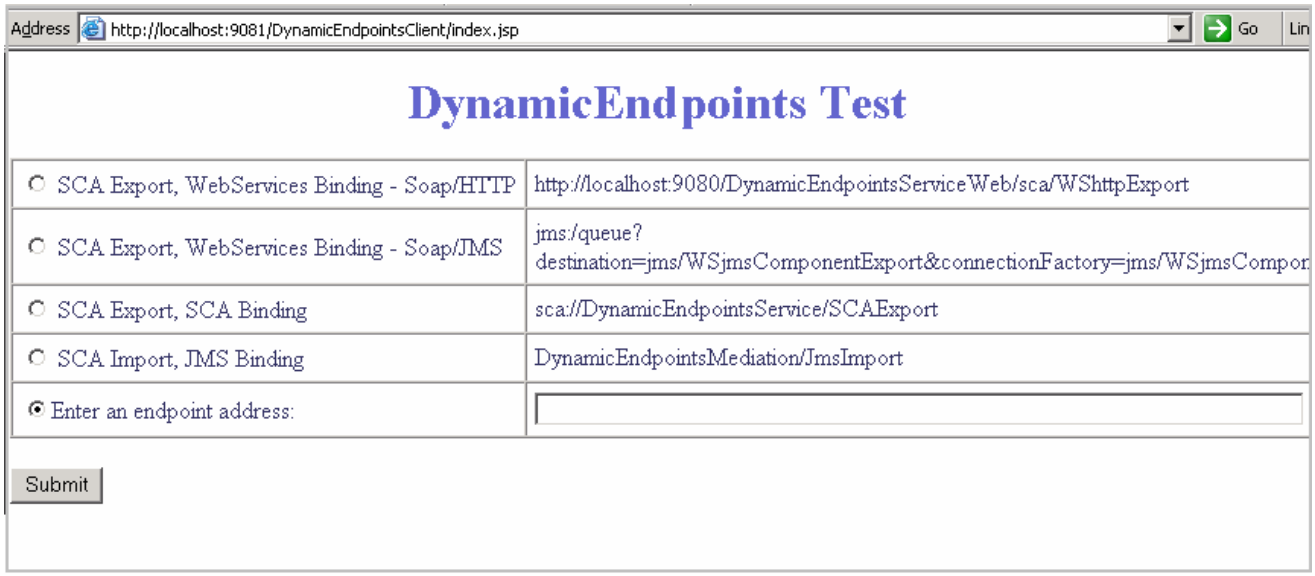
__ g. Enter http:// <HOST NAME>:<PORT>/DynamicEndpointsClient/index.jsp . Where **host name** is the fully qualified host name of the system where the WebSphere Enterprise Service Bus server is located. PORT is the **default_http** port of the WebSphere Enterprise Service Bus profile.

Ex: http://localhost:9080/DynamicEndpointsClient/index.jsp

Note: You can get the **default_http** port by going to **serverindex.xml** file in <WID_HOME>\pf\esb\config\cells\esbCell\nodes\esbNode. Where WID_HOME is the location where WebSphere Integration Developer is installed.

Ex: C:\WID602\pf\esb\config\cells\esbCell\nodes\esbNode

__ h. Select the radio button next to each binding type in turn, and press the **Submit** button



Note: The SOAP/HTTP endpoint address is the **default_http** port of the WebSphere Enterprise Service Bus profile. If other than default 9080, provide the correct HTTP port and use the text box and the radio button provided to request the URL. For example:

__ i. Verify that the endpoint requested matches the endpoint invoked

1) Invoking "**SCA Export, WebServices Binding – SOAP/HTTP**"

2) Invoking "**SCA Export, WebServices Binding – SOAP/JMS**"

3) Invoking "**SCA Export, SCA Binding**"

```
Requested URL --> sca://DynamicEndpointsService/SCAExport

Export/Component invoked --> SCAExport/SCAComponent
```

4) Invoking "**SCA Import, JMS Binding**"

```
Requested URL --> DynamicEndpointsMediation/JmsImport

Export/Component invoked --> JmsExport/JmsComponent
```

__ j. Close the browser

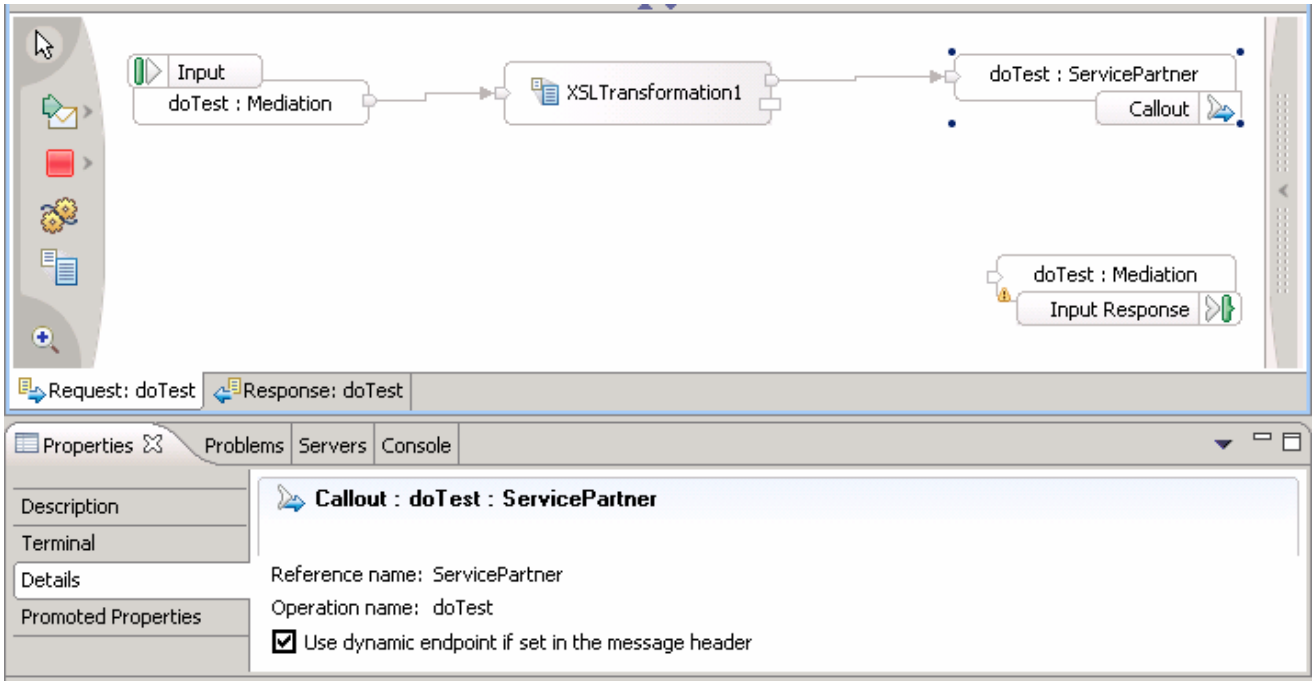
__ k. Remove the projects from the ESB server. To remove the projects, right-click over the ESB server in the Server view and select "**Add and Remove projects...**" from the context menu. From the Add and Remove dialog click over the **Remove All** and then click **Finish**

____ 4. Return to the Mediation Flow Editor and update the **Callout** property to disable the dynamic endpoints property.

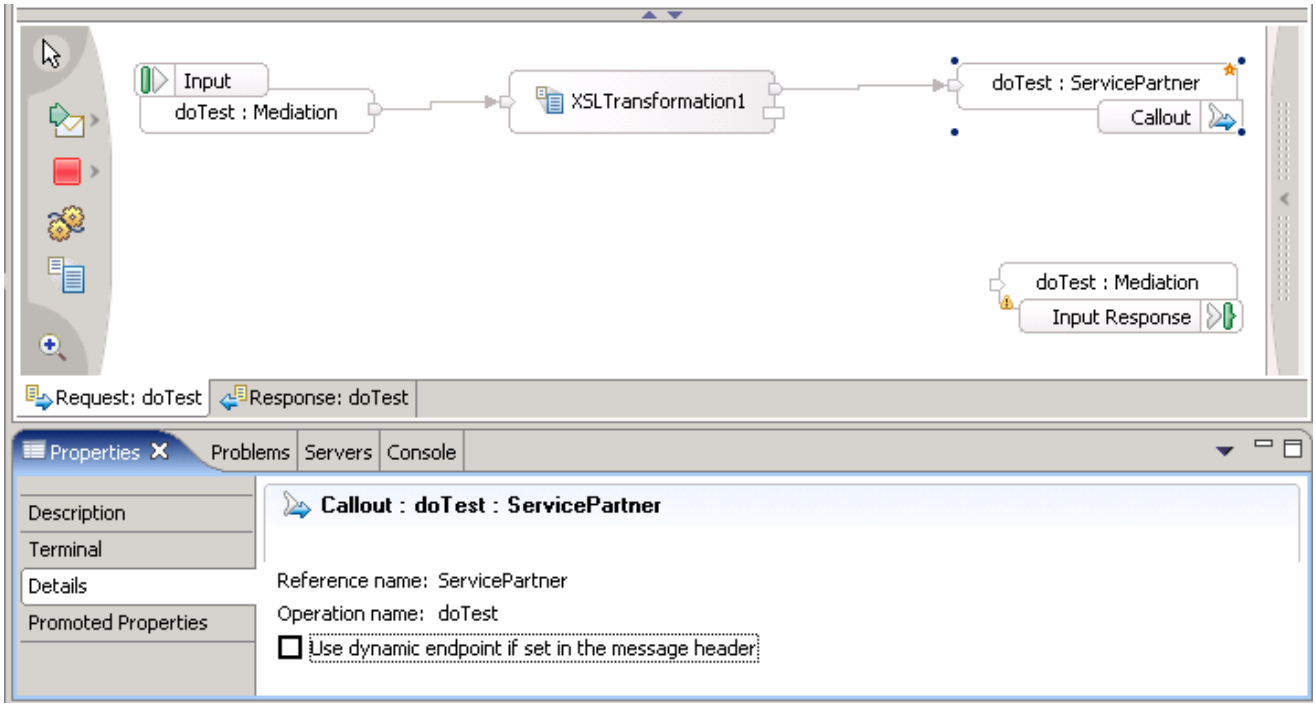
__ a. In the Business Integration view, expand **DynamicEndpointsMediation -> Mediation Logic ->Flows** and double click on **Mediation1** to open Mediation Flow Editor.

__ b. In the Mediation Flow Editor, click on the Mediation Flow line which will open the middle window

__ c. In the middle window, select the Request **Callout** node and select the **Details** tab under its properties view to modify the callout property



__ d. Unselect the check box next to "**Use dynamic endpoint if set in the message header"**



__ e. Save all work by choosing **File > Save All** or **Crtl + Shift + S**

__ f. Add all the projects again to run on the ESB server

__ g. Click on the Web browser icon to launch a browser in WebSphere Integration Developer

__ h. Enter http:// <HOSTNAME>:<PORT>/DynamicEndpointsClientWeb/index.jsp . Where hostname is the name of the system where the WebSphere Enterprise Service Bus server is located. Port is the **default_http** port of the WebSphere Enterprise Service Bus profile.

Ex: http://localhost:9080/DynamicEndpointsClientWeb/index.jsp

__ i. Select the radio button next to each binding type in turn, and press the **Submit** button

__ j. Verify that the requested endpoint invoked is **DefaultComponentExport** for each request

1) Invoking "**SCA Export, WebServices Binding - Soap/HTTP**"

```
Requested URL --> http://localhost:9081/DynamicEndpointsServiceWeb/sca/WShttpExport

Export/Component invoked --> DefaultComponentExport/DefaultComponent
```

2) Invoking "**SCA Export, WebServices Binding – Soap/JMS**"

```
Requested URL --> jms:/queue?
destination=jms/WSjmsComponentExport&connectionFactory=jms/WSjmsComponentExportQCF&

Export/Component invoked --> DefaultComponentExport/DefaultComponent
```

3) Invoking "**SCA Export, SCA Binding**"

```
Requested URL --> sca://DynamicEndpointsService/SCAExport

Export/Component invoked --> DefaultComponentExport/DefaultComponent
```

4) Invoking "**SCA Import, JMS Binding**"

```
Requested URL --> DynamicEndpointsMediation/JmsImport

Export/Component invoked --> DefaultComponentExport/DefaultComponent
```

__ k. Close browser.

__ l. Remove the projects from the ESB Server

____ 5. The exercise is complete

Part 5: Save the work and clean up server

____ 6. Export project as Project Interchange file

__ a. In WebSphere Integration Developer, Navigate to **File → Export**.

__ b. Select **Project Interchange**.

__ c. Out of all the projects listed, select only the following projects:

- DynamicEndpointsClient
- DynamicEndpointsLibrary
- DynamicEndpointsMediation
- DynamicEndpointsService
- Websphere_default_messaging_provider

__ d. Save in **C:/LabFiles602/WESB/DynamicEndpoints/**

__ e. Name the project interchange **WESB_DynamicEndpoints_Solution_PI.zip**

__ f. Click **Finish** to save the file

____ 7. Remove all the projects and **clean** up the ESB Server if not already done.

__ a. Right-click on WebSphere ESB Server V6.0 (once started) and select **Add and Remove projects...** from the context menu

__ b. Select **Remove-All** and click **Finish**

__ c. After the projects are removed, **stop** the WebSphere ESB Server V6.0

What you did in this exercise

In this lab, you were provided with an understanding of how to create a mediation module and mediation flow in WebSphere Integration Developer V6.0.2. A XSLT primitive was added to the Request and Response flow of the Mediation and the Service Message Object Header was mapped from source to target using the XSL editor. You also tested the application by deploying it to the integrated WebSphere Enterprise Service Bus test server with Callout property enabled and then disabled

Solution Instructions

____ 1. Import **Solution** Project Interchange file.

__ a. With a blank workspace in WebSphere Integration Developer, Go to **File → Import → Project Interchange**

__ b. Click on top Browse button and navigate to **C:/LabFiles602/WESB/DynamicEndpoints/WESB_DynamicEndpoints_Solution_PI.zip**

__ c. Select All Projects and click the **Finish** button. Ignore any warnings reflected in the Problems view

____ 2. OPTIONAL: If testing on a remote system, complete **3.i** and **3.j** in **Part 1: Prepare environment for the lab**

____ 3. Start with **Part 4: Test Dynamic End Points**

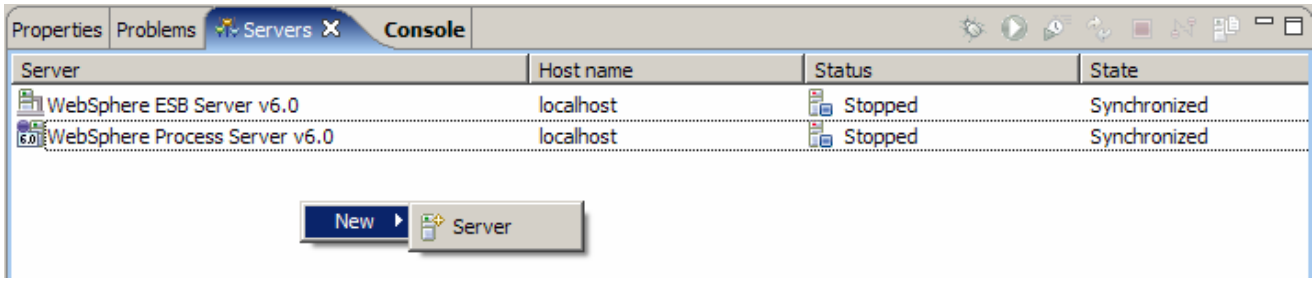
Task: Adding remote server to WebSphere Integration Developer test environment

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

____ 1. Create a new remote server

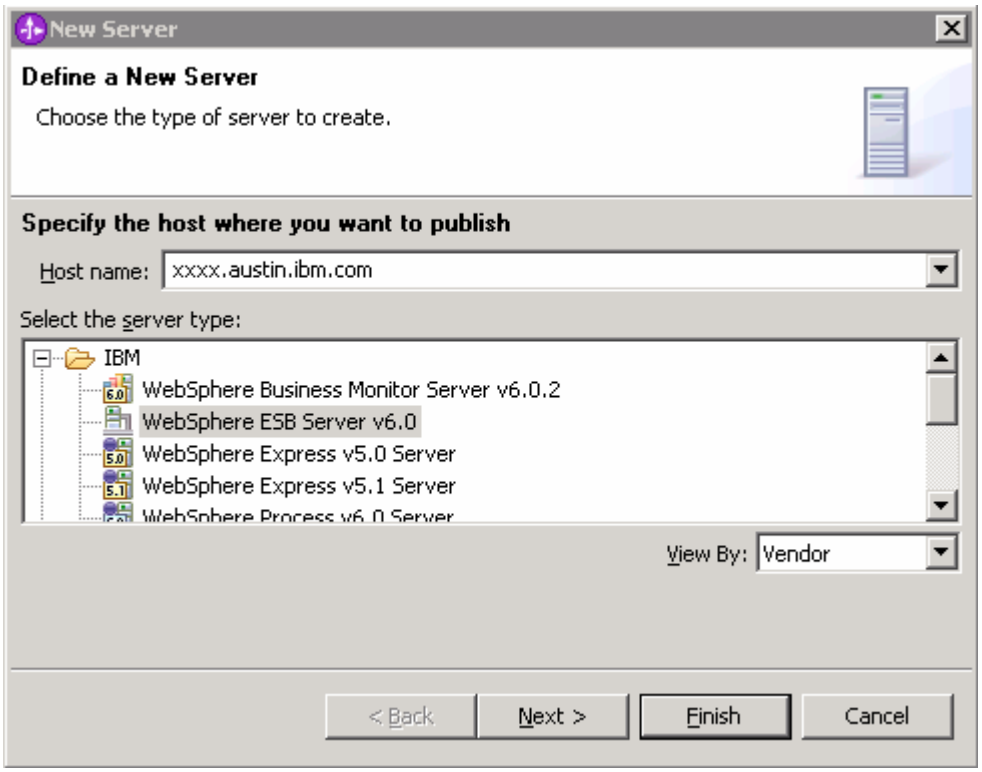
__ a. Right click on the background of the Servers view to access the pop-up menu

__ b. Select **New > Server**



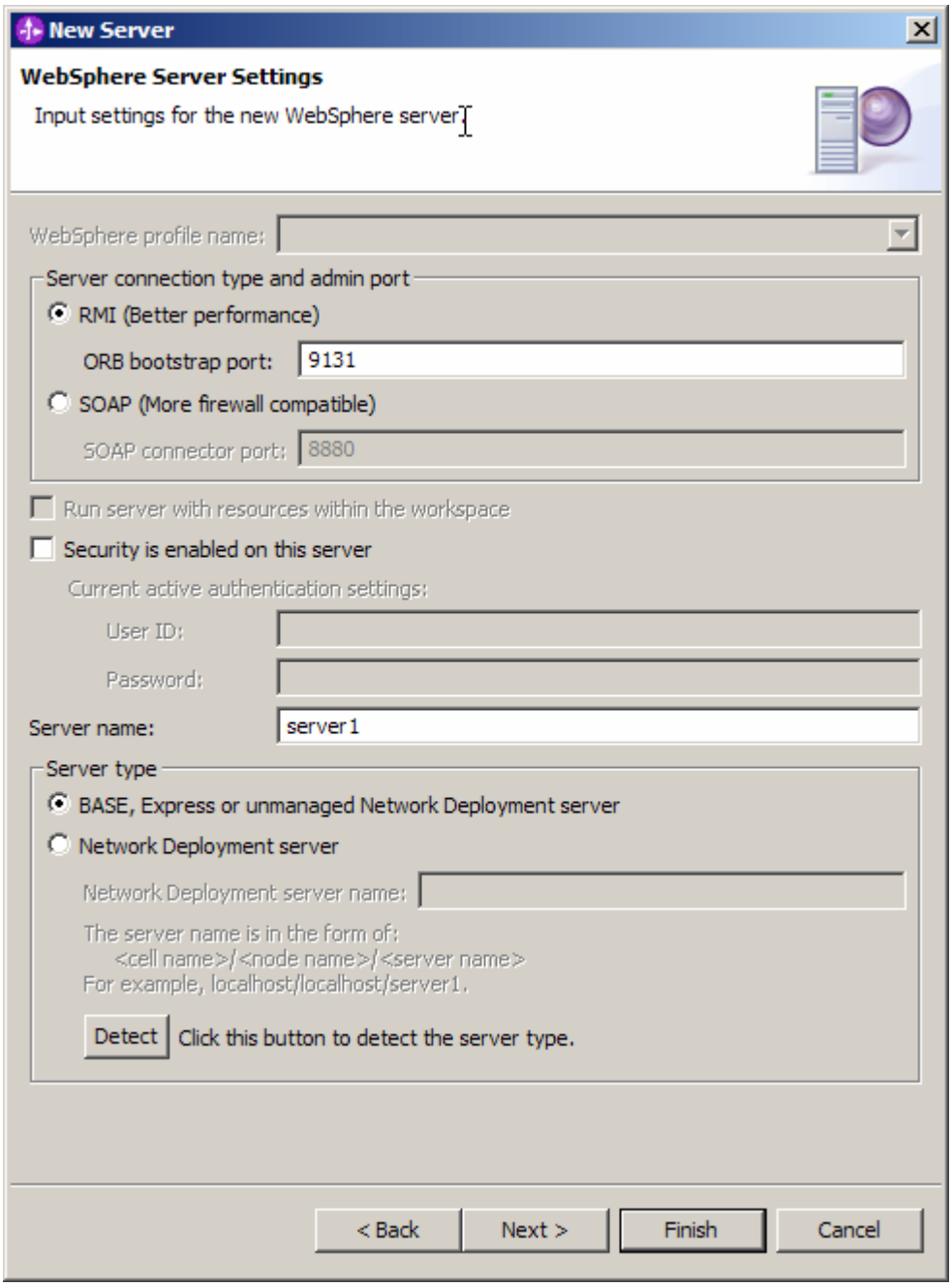
__ c. Specify hostname to the remote server, **<HOSTNAME>**

__ d. Ensure that '**WebSphere ESB 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

____ 2. 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>**'

User name: **<USERID>**

Password: **<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 cl1sr01 open for e-business; process id is 0000012000000002
```