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

WebSphere Enterprise Service Bus lab – Dynamic endpoints

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	3
Exercise instructions	6
Part 1: Prepare environment for the lab	7
Part 2: Create a mediation module	13
Part 3: Generate mediation flow implementation for the mediation	22
Part 4: Test dynamic end points	35
Part 5: Save the work and clean up server	42
What you did in this exercise	43
Solution Instructions	44
Task: Adding remote server to WebSphere Integration Developer test environment	45

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)

© Copyright IBM Corporation 2007. All rights reserved





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></wid_home>	Ex: C:\WID602	
<lab_files></lab_files>	C:\Labfiles602	/tmp/Labfiles602
<temp></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></server_name>	server1	cl1sr01	
<was_home></was_home>	C:\Program Files\IBM\WebSphere\App Server	/etc/cl1cell/AppServerNode1	
<hostname></hostname>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<port></port>	9080	9080	
<bootstrap_port></bootstrap_port>	2809	2809	
<telnet_port></telnet_port>	N/A	1023	
<profile_name></profile_name>	AppSrv01	default	
<userid></userid>	N/A	cl1admin	
<password></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 "<u>Task: Adding remote server to WebSphere Integration Developer test</u> <u>environment</u>".

Part 1: Prepare environment for the lab

In this section of the lab, all the projects that are part of **WESB_DynamicEndpoints_Pl.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

🕀 Workspace Launcher		x
Select a workspace		
IBM WebSphere Integration Developer stores your projects in a direct Select the workspace directory to use for this session.	tory called a workspace	
Workspace: C:\LabFiles602\WESB\DynamicEndpoints\workspace		Browse
	ОК	Cancel

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_Pl.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

Business Integration 🗙				' 🗖
	⊕	⊜	₽	•
New 🕨				
Op <u>e</u> n				
Сору				
Paste				
🗙 <u>D</u> elete				
i≥₁ <u>I</u> mport				
🛃 Exp <u>o</u> rt				
🐑 Re <u>f</u> resh				
P <u>r</u> operties				

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

👍 Import	×
Select Import a project and its dependent projects from a Zip file.	Ľ
Select an import source: File system Froject Interchange RAR file Server Configuration	
< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

- ___ 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**

🚯 Import Project II	nterchange Contents				x
Import Projects Import Projects from	a zip file.				
From zip file: Project location root:	C:\LabFiles602\WESB\Dyr C:\LabFiles602\WESB\Dyr	namicEndpoints namicEndpoints	\WESB_Dynamic	Endpoints_PI.zip 💌	Browse Browse
 ✓ (⇒ DynamicEndpoint) ✓ (⇒ DynamicEndpoint) ✓ (⇒ DynamicEndpoint) ✓ (⇒ websphere_doint) 	intsClient intsLibrary iintsService efault_messaging_provider				
Select All Deselect	t All Select Referenced]			
		< <u>B</u> ack	Next >	Einish	Cancel

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



 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.

Services
WShttpComponentExport_ServiceHttpService WShttpComponentExport_ServiceHttpPort Soap:address

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

Properties 🛛	Problems Servers Console Progress	
General	📄 address	
Documentation	Property	Value
	location	http://localhost:9080/Service455Web/sca/WShttpComponentExport

- 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.

🚯 Find/Replace	×
Eind: localho	st:9080 💌
Replace With: mvs22	2.rtp.raleigh.ibm.com 💌
Direction © F <u>o</u> rward © <u>B</u> ackward	Scope All Selected Lines
Options <u>C</u> ase Sensitive <u>W</u> hole Word Regular expression	Wrap Search
Find	Replace/Fin <u>d</u>
<u>R</u> eplace	Replace <u>A</u> ll
	Close

- 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)

Properties	Problems	👯 Servers 🗙	
Server			Host name
WebSp	here ESB S	erver v6.0	localhost

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**

🚯 New Mediation Module 🛛 🛛 🗙
Mediation Module
Create a new mediation module. A mediation module is a project that is used for development, version management, organizing resources, and deploying to the ESB runtime environment.
Module Name DynamicEndpointsMediation
Module Location
🔽 Use default
Directory: C:\RUSSworkSpace4\DynamicEndpointsMediation Browse,
Target Runtime: WebSphere ESB Server v6.0
Target Runtime: WebSphere ESB Server v6.0
Image: Target Runtime: WebSphere ESB Server v6.0 Image: Create mediation flow component Mediation modules can be deployed and run on WebSphere Enterprise Service Bus or WebSphere Process Server. They contain flows, which link together operations for modifying and routing messages between service consumers and service

____ 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

🚯 New Mediation Module 🔹 🕨	C
Select Required Libraries	
Select libraries containing re-usable resources such as interfaces, to be used by this module.	
Libraries	
DynamicEndpointsLibrary	
Module dependencies are primarily configured using the dependency	
editor.	
	_
< Back Mext > Finish Cancel	

- ____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

Dependencies: DynamicEndpointsMediation 🗙	
 Libraries Configure the required libraries. 	
amicEndpointsLibrary	Advanced:
Add Remove	
▶ Java	
J2EE	
Ordering	
Unresolved Projects	

- ___ 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

🔂 J2EE Project Selection
Select a project (? = any character, * = any String):
*
Contemporation and the second
OK Cancel

____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

式 *Dependencies: DynamicEndpointsMediation 🗙			
 Libraries Configure the required libraries. 			
2 DynamicEndpointsLibrary	Advanced: Deploy with Module		
Add Remove			
 ✓ J2EE Configure the dependent J2EE projects. The selected projects will be deployed as part of the module. 			
SphamicEndpointsClient Beging_provider	Advanced: On Build Path Deploy with Module		
Add Remove			

- ____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

۲
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 (E)

(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

➡ Stand-alone References	🗗 Mediation 1

- 2) Hover over the Stand-alone Reference or click on the Stand-alone Reference so the Add Reference icon appears. Click the icon (¹) 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

🚯 Add	Reference	×
?	There are one or more references on this component that are described by WSDL interfaces. It is simpler to develop Java clients if they use component references that are described as Java interfaces. Would you like to convert the WSDL interfaces used by this component's references so that they use Java interfaces?	
🗌 Rer	nember my decision and do not ask me again.	
	Yes Help	

- 5) Click NO to the Question on the pop-up dialog
- 6) The Stand-alone Reference must look as below:

Stand-alone References 1.1 Mediation1			
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			
 Right click over the import and select Rename from the context menu to change default import name from Import1 to WiredImport 			
Stand-alone References 1.1 (Mediation 1)			
 3) Hover over or click on WiredImport so the add interface icon appears. Click the icon (¹) to add an interface 4) Select Service from the list of matching interfaces 			
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			
 Right click over the import and select Rename from the context menu to change the default import name from Import1 to JmsImport 			
Stand-alone References 1.1 (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2			

诗 JmsImport

- 3) Hover over or click on the JmsImport so the add interface icon appears. Click the icon (
- 4) Select Service from the list of matching interfaces

Matching interfaces:	
Client Service	

- 5) Click OK
- 6) The Assembly Diagram looks like the one shown below:

Stand-alone References	Mediation1	1 🕞 WiredImport
		1 🕞 JmsImport

____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

Stand-alone References	Mediation 1	-•1 🕞 WiredImport
		(1) (1) JmsImport

- 3) Leave the JmsImport component alone
- ____g. Generate SCA bindings for the Imports

- 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:

Stand-alone References 1.1	📲 🗊 🐨 WiredImport
	1 🕵 JmsImport

- ____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

Properties 🗙	Problems Servers	
Description	🗟 Import: WiredImport (SCA Binding)	
Details	Module name:	
Binding	Export name: Browse	e

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

🔲 Properties 🗙	Problems Servers	
Description	🗟 Import: WiredImport (SCA Binding)	
Details	Module name: DynamicEndpointsService	
Binding	Export name: DefaultComponentExport	Browse

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**

IBM WebSphere Enterprise Service Bus 6.0.2 – Lab Exercise

- 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

Properties × Problem	s Servers	
Description	😪 Import: JmsImport (JMS Binding)	
Details	Connection JMS Destinations	
Binding	Send Destination Properties	
End-point configuration Method bindings Security attributes Summary	Select configuration view option: Specify JNDI name for pre-configured messaging provider resource Specify properties for configuring new messaging provider resource	
	JNDI Lookup Name: DynamicEndpointsService/JmsExport_RECEIVE_D	
	Receive Destination Properties	
	Callback Destination Properties	

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

🕕 🔁 Media		
	💛 Undo Update Destination JNDI Lookup Name	
	Nedo	
	Add	►
	Convert to Import	
	Generate Export	►
	Generate Implementation	
	Convert to Import Generate Export Generate Implementation	•

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

🚯 Generate Implementation	
Select the folder where the Mediatio	n Flow implementation will be generati
	•
1	
	OK Cancel
	OK Cancel

___ d. Click **OK**

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

🕄 *Assembly Diagram: DynamicEndpointsMediation 🦓 Mediation Flow Editor: Mediation1 🗙 🗖					
▼Operation connections 🖉 🔊 🧟 💥					
Select a source operation, connect it to one or more target operations, and define the mediation flow.					
(I) Client	Client ServicePartner				
🎲 doTest		🤯 doTest			
Mediation flow	Mediation flow				
Select a source operation to define the mediation flow.					

- 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

(I) Client	
👹 doTest	0
-	

_____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**

(I) Client	ServicePartner
🤯 doTest	 👹 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

😪 *Assembly Diagram: DynamicEndpointsMediation 🧖 🎭 *Mediation Flow Editor: Mediation	×	
•Operation connections Select a source operation, connect it to one or more target operations, and define the mediation flow.		
Client doTest	ServicePartner	
Input doTest : Client	doTest : ServicePartner Callout 🏠 doTest : Client Input Response 🔊	
€ ⊇ Request: doTest ↓ Response: doTest		

- _____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.

Input doTest : Client	doTest : ServicePartner Callout
	doTest : Client Input Response
Request: doTest	
Properties X Problems Servers	▼ □ □
Description Callout : doTest : ServicePartner Terminal Details Reference name: ServicePartner Operation name: doTest Operation name: doTest W Use dynamic endpoint if set in the message header	

- 5. Add a **XSL Transformation** primitive to the Request Mediation Flow diagram
 - ____ a. In the Mediation Flow Editor(middle), click on XSL Transformation icon (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

Input		doTest : ServicePartner
doTest : Client	🖻 📲 XSLTransformation1 🖁	Callout ≽

____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

Properties 🗙 Problem	ems Servers	- 1	- 0
Description	Transformation : XSLT ransformation1		1-
Terminal	🐼 Mapping file: cannot be empty.		
Details	Root: <not specified=""></not>		
Promoted Properties	Mapping file: * <select a="" file="" mapping=""> Browse Edit,</select>	ew]
			•

- ____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

🚯 New XSLT Mapping	×
Specify Message Types	
Select Input and Output Message Type	5
Message Root:	
Input Message Body: doTestRequestMsg	Browse
Output Message Body: doTestRequestMsg	Browse
Defined Contexts	
Correlation Context: (1) No correlation context is set fo Contexts can be set on the inp	r this flow. ut node of the flow.
Transient Context: (A) No transient context is set for Contexts can be set on the inp	this flow. ut node of the flow.
< Back Next > Ein	ish Cancel

___ d. Click Finish

____e. The XSL Transformation mapping editor opens

🕄 *Assembly Diagram: DynamicEndpointsMedi	🌼 *Mediation Fl	low Editor: Mediation1	XSLTransformation1_re	q_1.xmx 🗙 🔲
🗴 Source 🕹 个	- ♦ 🕞 🕀 🔻	🗴 Target	ት 🕂 🕂	🕆 🗔 🌾 💌
⊡ 🕅 ⁴ smo È 🖻 tns:smo		⊡ 🖈 Èsmo È € tns:smo		
Noterview				😈 🚯 🖝 👻
Target	Source		Applied Function/Grouping	
🗴 smo	🗴 smo			

_____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:
 - 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:

Target	Source
- 🗴 smo	🗴 smo
e tns:smo	
e headers	
SMOHeader [01]	SMOHeader [01]
e MessageUUID	e MessageUUID
e Version	e Version
e Version	e Version
e Release	e Release
e Modification	e Modification
e MessageType [01]	MessageType [01]
e Operation [01]	Coperation [01]
e Action [01]	C Action [01]
e Target [01]	
e address	endpoint [01]
e body [01]	
doTest	
- e inputDoTest	
endpoint [01]	endpoint [01]

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:

Properties 🛛 Probl	ems Servers	, – –
Description	Transformation : XSLTransformation1	1
Details Promoted Properties	Root: /	
	Associated X5L: xslt/XSLTransformation1_req_1.xsl	
		-

____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:

😵 *Assembly Diagram: DynamicEndpointsMediation 🦷 🦓 *Mediation Flow Editor: Me	diation1 🗙 🖉
🕶 Operation connections 🛛 🔊 💀 🗶	
Select a source operation, connect it to one or more target operations, and define the	mediation flow.
(I) Client	ServicePartner
🤯 doTest	📷 🕷 doTest
▲ ▼	1
Callout Response doTest : ServicePartner	doTest : Client
	<
Request: doTest Response: doTest	
a. In the Mediation Flow Editor(middle), click on XSL XSL Transformation primitive from the pallet on left-h between the Callout Response node and the Input R	and side of view and drop it into the canvas esponse Node
doTest : ServicePartner	doTest : Client
b. Hover the mouse over Callout Response node's our appears to the input terminal of the XSL Transformat	tput terminal and drag the handle that ion primitive, XSL Transformation1
doTest : ServicePartner	on1 doTest : Client Input Response
c. Hover the mouse over XSL Transformation primitive, and drag the handle that appears to the input termina	XSL Transformation1's output terminal al of the Input Response node
doTest : ServicePartner	doTest : Client
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

🔲 Properties 🗶 🛛 Prob	ems Servers		•	
Description Terminal Details Promoted Properties	Image: State Stat	Browse Edit Ne	ew	

- ____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

🚯 New XSLT Mapping	×
Specify Message Types	
Select Input and Output Message Type	5
Message Root:	
Input Message Body: doTestRequestMsg	Browse
Output Message Body: doTestRequestMsg	Browse
Defined Contexts	r this flow
Correlation Context: (1) Contexts can be set on the inp	ut node of the flow.
Transient Context: (A) No transient context is set for Contexts can be set on the inp	this flow. ut node of the flow.
< Back Next > Ein	ish Cancel

- ___ 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:

Properties 🛛 Probl	ems Servers 🗸 🗸	- 8
Description Terminal	Transformation : XSLTransformation1	
Details Promoted Properties	Root: / Mapping file: * xslt/XSLTransformation1_res_1.xmx Associated XSL: xslt/XSLTransformation1_res_1.xsl]

____10. **Close** the Mediation Flow Editor

Part 4: Test dynamic end points

1.

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

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)

Start WebSphere ESB Server and add modules to server

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

Status	State
Contraction of the second s	
Stopped	Synchronized
	ES 200thed

- ____ 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:

🚯 Add and Remove Projects		×
Add and Remove Projects Modify the projects that are configured o	in the server	
Move projects to the right to configure the	m on the server	
<u>A</u> vailable projects:	<u>C</u> onfigured projects:	
⊕ ⊡ DynamicEndpointsServiceApp ⊕ ⊡ DynamicEndpointsMediationApp	Add >	
	<	Cancel

- ____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

vailable command groups: Menubar Details:	Toolbar Details:
Java Open Actions Java Search JUnit Launch Open External Files Profile Proxy Tools ReportingToolbarLabel Resource Navigation SampleGalleryActionSetLabel Search Stearch Stearch Software Updates Software Updates Software Updates Software Updates Software Updates	Navigate toolbar

___ 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

Address 🚳 http://localhost:9081/DynamicEndpointsClient/index.jsp	💌 🄁 Go 🛛 Lin		
DynamicEndpoints Test			
O SCA Export, WebServices Binding - Soap/HTTP	http://localhost:9080/DynamicEndpointsServiceWeb/sca/WShttpExport		
O SCA Export, WebServices Binding - Soap/JMS	jms:/queue? destination=jms/WSjmsComponentExport&connectionFactory=jms/WSjmsCompon		
O SCA Export, SCA Binding	sca://DynamicEndpointsService/SCAExport		
O SCA Import, JMS Binding	DynamicEndpointsMediation/JmsImport		
• Enter an endpoint address:			
Submit			

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:

• Enter an endpoint address:	http://localhost:9081/DynamicEndpointsServiceWeb/sca/WShttpExport
Submit	

_____i. Verify that the endpoint requested matches the endpoint invoked

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

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

Export/Component invoked --> WShttpExport/WShttpComponent

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

Requested URL --> jms:/queue?

destination=jms/WSjmsComponentExport&connectionFactory=jms/WSjmsComponentExportQCF&

Export/Component invoked --> WSjmsExport/WSjmsComponent

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

Input doTest : f	lediation	XSLTransformation1	doTest : ServicePartner Callout
Request: doTest 🖓	esponse: doTest		
Properties 🕅 Proble	ms Servers Console		▼ □ □
Description Terminal Details Promoted Properties	Callout : doTest : ServicePartment Reference name: ServicePartment Operation name: doTest	vicePartner er : in the message header	

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

	A V		
Input doTest : M	lediation	1 doTest : ServicePartner	
			<
•		doTest : Mediation	
Request: doTest 🖓 Re	esponse: doTest		
Properties X Proble	ms Servers Console	~	
Description Terminal	allout : doTest : ServicePartner ≽		
Details	Reference name: ServicePartner		
Promoted Properties	Operation name: doTest		
	Use dynamic endpoint if set in the message heade	er	

- ____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 <a href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://<href="http://http:/http://http:/http:/http://http://http://http://http://http://http

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/WSjmsComponentExportQCF8

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.
- ___I. 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 \rightarrow 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

Properties Problems 🐼 Servers 🗙 Console		🌣 🕥 🤣	🍫 🔲 🙌 💷 🗖
Server	Host name	Status	State
HebSphere ESB Server v6.0	localhost	🚡 Stopped	Synchronized
WebSphere Process Server v6.0	localhost	🖥 Stopped	Synchronized
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>**)

🚯 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	
C 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, localbost/localbost/server1</server></node></cell>	
Detect Click this button to detect the server type.	
	Consul 1
< Back Next > Finish	Cancel

- ___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 cllsr01 open for e-business; process id is 0000012000000002