# WebSphere Enterprise Service Bus lab – Event emitter primitive

What this exercise is about	2
Lab requirements	2
What you should be able to do	2
Introduction	3
Exercise instructions	5
Part 1: Prepare environment for the lab	6
Part 2: Create mediation module and flow	9
Part 3: Deploy the modules and configure Web services explorer	26
Part 4: Event generation test A and B using the Web services explorer	34
Part 5: Event generation test C using the test client	44
Part 6: Save the work and clean up server	50
What you did in this exercise	51
Solution instructions	52
Task: Adding remote server to WebSphere Integration Developer test environment	53

### What this exercise is about

The objective of this lab is to demonstrate the ability to produce different events based on the event emitter primitive configuration. It focuses particularly on the event label and root properties which define the different event types at runtime.

### 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 ESB 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 a Mediation Module and a Mediation Flow
- Visually compose the Mediation Flow that was created
- Run the Mediation Module on the WebSphere Process Server 6.0 test server
- Test the Event Emitter primitive to produce events using the Web Service Explorer
- Also do the same test using a Test Client

## Introduction

The scenario in this lab demonstrates the ability to produce different events based on the event emitter primitive configuration. It focuses particularly on the event label and the root properties which define the different event types at runtime.

The Mediation flow represented in the diagram below; has the request flow with two Event Emitter primitives and the response flow with one Event Emitter primitive.



The following are the permutations which ensure that the tools generate appropriate default event labels for request and response flows. At runtime the mediation flow should emit three different types of events testing the runtimes ability to specify the event's extension name and content. The message filter primitive filters on the age element in the CustomerDetails business object. If the value of 'age' is equal to or greater than 16, event emitter 1 will be run in the request flow; otherwise event emitter 2 will be run.

Event Emitter 1 (Request Flow)				
Event Label	<default by="" generated="" the="" tools=""></default>			
Root	<unspecified></unspecified>			
Transaction Mode	Default			
Event Emitter 2 (Request Flow)				
Event Label	"JuniorAccount"			
Root	/body			
Transaction Mode	Default			
Event Emitter 1 (Response Flow)				
Event Label	<default by="" generated="" the="" tools=""></default>			
Root	"/body/getCustomerID/input1/customerID"			
Transaction Mode	Default			

*Note*: In this lab, the mediation flow contains only event emitter primitives and one message filter primitive but in true customer scenarios it is expected that the event emitter will be used in combination with other primitive types. However, as each mediation primitive is stateless, the addition of other primitives to this scenario would not add any additional test execution paths for the event emitter line item.

## **Exercise instructions**

Some instructions in this lab may be Windows<sup>®</sup> 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 <sup>®</sup> /UNIX <sup>®</sup> Location
<wid_home></wid_home>	C:\WID602	
<lab_files></lab_files>	C:\Labfiles602	/tmp/Labfiles602

**Windows users' note**: When directory locations are passed as parameters to a Java<sup>™</sup> 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	
<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 "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\_EventEmitterPrimitive\_PI.zip** project interchange file are imported into a new workspace.

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

Workspace I	auncher			X
Select a wo	rkspace			
Select the w	ere Integration Developer stores your p orkspace directory to use for this sessio	n.	called a workspace	э.
<u>W</u> orkspace:	C:\LabFiles602\WESB\EventEmitterPrim	nitive\workspace	<u> </u>	Browse
Use this a	is the default and do not ask again			
			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 window

- \_\_\_\_3. Import the Project Interchange file, **WESB\_EventEmitterPrimitive\_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

Business Integration 🗙				• 🛛
	⊕	⊜	£₽}	•
New 🕨				
Op <u>e</u> n				
Сору				
Paste				
🗶 Delete				
<u>i</u> mport				
🛃 Exp <u>o</u> rt				
🔊 Refresh				
P <u>r</u> operties				
				1

\_\_\_\_ 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 File system Froject Interchange RAR file Server Configuration	
< Back Next > Einish	Cancel

\_\_\_ d. Click Next

\_\_\_\_e. Click the **Browse** button for "**From zip file**" to navigate for the Project interchange file, WESB\_EventEmitterPrimitive\_PI.zip

👍 Import Project Interchange Contents	×
Import Projects Import Projects from a zip file.	
From zip file:       C:\LabFiles602\WESB\EventEmitterPrimitive\WESB_EventEmitterPrimitive_PI.zip         Project location root:       C:\LabFiles602\WESB\EventEmitterPrimitive\workspace         Image: CustomerIDModule       Image: CustomerIDModule	Browse
Select All Deselect All Select Referenced	
< Back Next > Finish	Cancel

- \_\_\_\_f. Click the Select All button to ensure all projects listed are selected
  - Projects listed are : CustomerIDModule and CustomerLibrary
- \_\_\_\_g. Click the Finish button (projects will be imported and auto-build will run)
- \_\_\_ h. Verify that the CustomerIDModule and CustomerLibrary modules are listed in the Business Integration view



\_\_\_\_4. Verify that the WebSphere ESB Server v6.0 is listed in the **Servers** view

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

### Part 2: Create mediation module and flow

In this section of the lab, a new mediation module and a Mediation Flow are created. The request and response flows are created for the Mediation flow. Further the Mediation Module is visually composed using the assembly editor.

- \_\_\_\_\_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

E Busine	ess Integration 🗙	
	# 🗁 🖻	\$ <b>-</b>
	ustomerIDModule ustomerLibrary	
	Ne <u>w</u>	Project
	Open Assembly Diagram	
	Open Deployment Editor	💯 Mediation Module

- \_\_\_\_b. In the New Mediation Module window, type the Module Name as EventEmitterTest
- \_\_\_\_ c. Verify that the target runtime is the WebSphere ESB Server v6.0 and clear the "Create mediation flow component" check box

Bew 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 N <u>a</u> me: EventEmitterTest
Module Location
✓ Use default
Directory; C:\LabFiles602\WESB\EventEmitterPrimitive\workspace\Event Browse
Iarget Runtime:         WebSphere ESB Server v6.0
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 providers.
< <u>Back</u> <u>N</u> ext > <u>Finish</u> Cancel

#### \_\_\_ d. Click Next

\_\_\_\_e. In the Select Required Libraries wizard, select CustomerLibrary

🚯 New Mediation N	1odule 🛛			×
Select Required L	ibraries			
Select libraries conta be used by this mod	aining re-usal ule.	ble resources	such as interface	is, to
Libraries	.]			
	< Back	Nexts	Finish	Cancel

- \_\_\_\_f. Click Finish. A mediation module called EventEmitterTest is created
- 2. Create a Mediation Flow by completing the following steps:
  - \_\_\_\_a. In the Business Integration View, right-click on EventEmitterTest Mediation Module to see the context menu and select New → Mediation Flow

🔠 Business Integra	tion 🗙 🗧	' D)(	
	# 🕞 🖨 🕏	-	
🕀 🔁 CustomerIDf	Module		
🕀 🔂 CustomerLib	rary		
	New		
	Open Assembly Diagram		
	Open Deployment Editor		😢 Mediation Module
	Open Dependencies		😤 Module
	Show Files		😭 Library
	🛅 Сору		📑 Business Object
	Paste		🕸 Custom Visual Snippet
	💢 <u>D</u> elete		🛗 Enterprise Data Discovery
	Refactor	•	😭 Enterprise Service Discovery
	nan Import		🖹 Event Definition
	Export		🗊 Interface
			💇 Mediation Flow

\_\_\_\_b. In the New Mediation Flow window, enter the Name as EmitterTestFlow

🚯 New Media	tion Flow	×
Create a new	mediation flo <del>w</del>	ſŊĴ
<u>M</u> odule: Name <u>s</u> pace: F <u>o</u> lder: N <u>a</u> me:	EventEmitterTest       New         http://EventEmitterTest/EmitterTestFlow       Default         Browse       Browse	
	< Back Next > Einish	Cancel

\_\_\_ c. Click Next

\_\_\_\_d. In the following window (creating a new interface), add the source and target interfaces

1) Click the Add button next to the Source interfaces text area and select CustomerDetails from the Interface Selection pop-up window

🕕 Ne	🔂 Interface Selection	X
Creal Sele	Filter by interface or qualifier (? = any character, $*$ = any String):	<b>•</b>
	Matching interfaces:	
	CustomerDetails	
		<u>A</u> dd <u>R</u> emove
	Qualifier: ①http://CustomerLibrary/CustomerDetails - CustomerLibrary/C	A <u>d</u> d R <u>e</u> move
	OK Cancel	
-		Cancel

- 2) Click **OK** over the **Interface Selection** pop-up window
- 3) Similarly, click the **Add** button next to the **Target interfaces** text area and select **CustomerDetails** from the **Interface Selection** pop-up window
- 4) Click **OK** over the **Interface Selection** pop-up window
- 5) The added source and target interfaces panel must like the one below:

	×
Creating a new interface Select source and target interfaces.	Į∑Ĵ
Source interfaces	<u>A</u> dd <u>R</u> emove
Target interfaces CustomerDetails - /CustomerLibrary/CustomerDetails.wsdl	A <u>d</u> d R <u>e</u> move
< <u>B</u> ack <u>M</u> ext > <u>F</u> inish	Cancel

6) Click Finish

\_\_\_\_e. A new Mediation Flow, that is, **EmitterTestFlow** is created for the **EventEmitterTest** Mediation Module as shown below:



\_\_ f. Upon creation of the **EmitterTestFlow** Mediation Flow, it is opened in the Mediation Flow Editor as shown in the diagram below:

Mediation Flow Editor: EmitterTestFlow S					
•Operation connections	🕶 Operation connections 🖉 💭 💭 💥				
Select a source operation, connect it to one	operations, and define the mediation flow.				
	CustomerDetailsPartner				
🤯 getCustomerID	🤯 getCustomerID				

#### Create a request flow:

To create the request flow, complete the following steps:

- 3. In the EmitterTestFlow's mediation flow editor, connect the source to target operations in the Operation Connections view
  - \_\_\_\_a. Click anywhere on the source operation, **CustomerDetails/getCustomerID** on the left-hand side of the Operation Connections view

<ol> <li>CustomerDetails</li> </ol>	
octomerID	

\_\_\_\_b. Drag to the target operation, **CutomerDetailsPartner/getCustomerID** on the right-hand side of the Operation Connections view and release the mouse click

*Alternative*: Right-click on the source component, **CustomerDetails/getCustomerID** and select **Create** an operation connection and then click on the target operation, **CutomerDetailsPartner/getCustomerID** 

(1) CustomerDetails	🔁 CustomerDetailsPartner
👹 getCustomerID	 👹 getCustomerID

- \_\_\_\_ c. Click on the black line (wire) to view the Mediation Flow View and ensure the Request tab is selected to build the Request flow
- \_\_\_\_\_4. Add an Message Filter primitive to the Request Mediation Flow diagram canvas
  - \_\_\_\_a. In the **Mediation Flow Editor**(middle window), click on the **Message Filter** icon (-) to select the Message Filter primitive from the pallet tray on left-hand side of view and drop it into the canvas between the Input Request node and the Callout Request Node

Note: Click Down arrow of the pallet tray to scroll and locate the Message Filter icon if not visible.

getCustomerID : CustomerD	MessageFilter1	getCustomerID : CustomerD Callout
b. Hover the mouse ov input terminal of the	er the <b>Input</b> node's output term Message Filter primitive, <b>Mess</b>	ninal and drag the handle that appears, to the <b>ageFilter1</b>
getCustomerID : CustomerD	••	getCustomerID : CustomerD

	Add a connection to an input terminal	
getCustomerID : CustomerD	MessageFilter1	getCustomerID : CustomerD Callout

\_\_\_\_ c. Right Click on the Message Filter Primitive, **MessageFilter1** and select **Add Output Terminal** from the context menu

MessageFilter1		
•	💛 Undo Add Link	
	₩ <u>R</u> edo	
	Select All	
	X Delete	
	💽 Zoom <u>I</u> n	Ctrl+=
	⊇ Zoom <u>O</u> ut	Ctrl+-
	Add Output Terminal	

#### \_\_\_\_d. Enter ageBelow16 for the Terminal name field

🚯 New Dynamic 1	erminal
Create a new dyr	amic terminal
Create a new dyna	amic terminal from a list of pre-defined terminal types.
Terminal type	match
Terminal name	ageBelow16
Terminal description	On successful processing, if the input message content matches an XPath patternList[i], the input message is propagated unchanged to the output terminal with the name
	OK Cancel

\_\_\_e. Click OK

\_\_\_\_f. The Message Filter primitive must look like the one below with a new Output terminal added:



\_\_\_\_g. Select the Message Filter primitive, MessageFilter1 and choose Details under its properties view

Properties 🗙 Probl	ems Servers		▼ - □
Description	🗯 Message Filter : MessageFilter1		<b>_</b>
Terminal			
Details	Distribution mode: First		<b>•</b>
Promoted Properties	Filters:		
	Pattern	Terminal name	Add
			Edit,

- \_\_\_\_h. Click the Add button for Filters
- \_\_\_\_i. The Add/Edit properties window opens

🕂 Add/Edit	x
Add/Edit properties	
😣 Pattern: cannot be empty.	
Pattern: * Terminal name:* ageBelow16	Custom XPath
	Einish Cancel

- \_\_\_\_j. From the Add/Edit properties window, click the Custom XPath button to set the Pattern. The XPath Expression Builder window opens
- \_\_\_\_k. In the **XPath Expression Builder** window, navigate to the **age** element (ServiceMessageObject> body>getCustomerID>input1>age) in the SMO body, and alter the XPath to **/body/getCustomerID/input1/age<16** as shown below:

*Note*: To Alter the XPath location, type <16 next to */body/getCustomerID/input1/age* for the XPath Location field.

🚯 XPath Express	sion Builder				×
Build the XPath Select the target or select Overrid	<b>Expression</b> from the Schema Viewer a e to manually enter an XP	and optionally defin ath expression	e a conditior	١,	55
Schema Viewer:					
ServiceMa	essageObject : PARENT ContextType HeadersType tCustomerIDRequestMsg ustomerID : _getCustomer put1 : CustomerDetails f_name : string l_name : string age : nillable:int address : Address	'IDParameters_			
XPath Location:	/body/getCustomerID/in	put1/age<16 🔶			
Condition:	(Location)		= (Value)	į	
Full XPath Expres	ssion:				
Override	perID/ipput1/age<16				
				OK	Cancel

\_\_\_I. Now select the check box next to **Override** under the Full XPath Expression section as shown below:

🕭 XPath Express	ion Builder					×
Build the XPath Select the target or select Override	<b>Expression</b> from the Schema Viewer a e to manually enter an XP	and optionally defin ath expression	ead	ondition,		55
Schema Viewer: ServiceMa Context : ServiceMa Context : ServiceMa beders : ServiceMa Service	essageObject : PARENT ContextType HeadersType tCustomerIDRequestMsg istomerID : _getCustomer put1 : CustomerDetails f_name : string l_name : string age : nillable:int address : Address	'IDParameters_				
XPath Location:						
Condition:	(Location)		=	(Value)		
Full XPath Expres	sion:					
(Verride)	erID/ipput1/age<16					
/body/geredston	ici Ioyinipuci/age<10					
					ОК	Cancel

\_\_\_ m. Click **OK** 

🚯 Add/Edit			×
Add/Edit propertie	\$		
Specify the properti	es.		
Pattern: *	/body/getCustomerID/input1/age<16	Custom XPath	
Terminal name:*	ageBelow16	•	
		<u>Einish</u> Cancel	

#### \_\_\_ n. Click Finish over the Add/Edit properties

\_\_\_\_\_o. The Message Filter Primitive Filter added must look as shown below:

💷 Properties 🗙 🛛 Proble	ems Servers Console		▼ □ □
Description	🗯 Message Filter : MessageFilter1		-
Terminal			
Details	Distribution mode: First		
Promoted Properties	Filters:		
	Pattern	Terminal name	Add
	/body/getCustomerID/input1/age<16	ageBelow16	

\_\_\_\_\_p. In the **Mediation Flow Editor**(middle window), click on the **Event Emitter** icon (<sup>1</sup>) to select the Event Emitter primitive from the pallet on left-hand side of view and drop **two** of them into the canvas between the, **MessageFilter1** and the Callout Request Node as shown below:

Input getCustomerID : CustomerD	EventEmitter1	getCustomerID : CustomerD Callout
MessageFilter1	EventEmitter2	

\_\_\_\_q. Wire the Message Filter primitive, MessageFilter1 and the Event Emitter primitives, EventEmitter1 and EventEmitter2 as shown below:

*Note*: The **ageBelow16** output terminal of the Message Filter is wired to the input terminal of **EventEmitter2** primitive. The **default** output terminal is wired to the input terminal of **EventEmitter1** primitive.

getCustomerID : CustomerD	MessageFilter1	EventEmitter1	getCustomerID : CustomerD Callout
l	ageBelow16 type:getCu:		

\_\_\_\_\_r. Also wire the output terminals of the Event Emitter primitives, **EventEmitter1** and **EventEmitter2** to the input terminal of the Request **Callout** node as shown below:

getCustomerID : CustomerD		EventEmitter1	getCustomerID : CustomerD Callout
	MessageFilter1	EventEmitter2	

\_\_\_\_s. Select the Event Emitter primitive, **EventEmitter1** and choose **Details** under its properties view. The property values for this primitive are the default values and should match the diagram below:

Properties 🕅 Probl	ems Servers		▼ - □
Description	🖄 Event Emitte	er : EventEmitter1	
Terminal			
Details	Label: *	EventEmitterTest_EmitterTestFlow_EventEmitter1_Req	
Promoted Properties	Root:	<exclude content="" data="" event="" from="" message=""></exclude>	Custom XPath
	Transaction mode:	Default 💌	

- \_\_\_\_t. Select the Event Emitter primitive, EventEmitter2 and choose Details under its properties view
  - 1) Enter JuniorAccount for the event Label
  - 2) Select /body from the Root drop down list
  - 3) Leave the Transaction mode as default with the value, Default

🔲 Properties 🔀 🛛 Proble	ems Servers		▼ □ □
Description	🖹 Event	Emitter : EventEmitter2	
Terminal			
Details	Label:	* JuniorAccount	
Promoted Properties	Root:	/body	Custom XPath
	Transaction r	mode: Default	•

\_\_\_\_\_u. Save all work by choosing File > Save All or Crtl + Shift + S

#### **Create a Response Flow:**

- 5. Create the Request Flow by completing the following steps:
  - \_\_\_\_a. While still in Mediation Flow Editor(middle section), select the **Response tab**
  - \_\_\_\_b. In the **Mediation Flow Editor**(middle), click on the **Event Emitter** icon (<sup>1</sup>/<sub>2</sub>) to select the Event Emitter primitive from the pallet on left-hand side of view and drop into the canvas as shown below

getCustomerID : CustomerD	EventEmitter1	getCustomerID : CustomerD Input Response
c. Wire the Event Emitter primit	ive, EventEmitter1 as shown bel	ow: getCustomerID : CustomerD Input Response 🔊

\_\_\_\_d. Select the Event Emitter primitive, **EventEmitter1** in the Response Flow and choose **Details** under its properties view

Properties 🛛 Proble	ems Servers 🗸 🗸 🗖
Description	Event Emitter : EventEmitter1
Terminal	
Details	Label: * EventEmitterTest_EmitterTestFlow_EventEmitter1_Res
Promoted Properties	Root: <a>  </a>
	Transaction mode: Default

- \_\_\_\_e. Click the **Custom XPath** button for the **Root** property. The **XPath Expression Builder** window opens
- \_\_\_\_\_f. In the **XPath Expression Builder** window, navigate to the **customerID** element; select it within the SMO body

🚯 XPath Express	sion Builder
Build the XPath I Select the target f or select Override	Expression from the Schema Viewer and optionally define a condition, e to manually enter an XPath expression
Schema Viewer:	
ServiceMe	essageObject : PARENT ContextType HeadersType tCustomerIDResponseMsg ustomerIDResponse : _getCustomerIDResult_ ustomerID : string
	/bodu/getCustomerIDDecoopse/sustamerID
XPath Location:	
Condition:	
Full XPath Express	ssion:
/body/getCustom	nerIDResponse/customerID
	OK Cancel

#### \_\_\_ g. Click OK

\_\_\_\_h. Leave all the other properties as default values. The properties view for the **EventEmitter1** primitive of the Response Flow must look like the diagram below:

🔲 Properties 🗶 🛛 Probl	rms Servers 🗸 🗸 🗖
Description	Event Emitter : EventEmitter1
Terminal	
Details	Label: * EventEmitterTest_EmitterTestFlow_EventEmitter1_Res
Promoted Properties	Root: /body/getCustomerIDResponse/customerID Custom XPath
	Transaction mode: Default

\_\_\_\_\_ i. Save all work by choosing File > Save All or Crtl + Shift + S

#### Visually compose the Mediation Module:

- \_ 6. To Visually compose the Mediation Module, complete the following steps:
  - \_\_\_\_a. In the Business Integration view's tree, expand the **EventEmitterTest** mediation module and double-click the mediation module assembly ( B Assembly Diagram ) to open it with the assembly editor
  - \_\_\_\_ b. An empty assembly editor for the EventEmitterTest mediation module is opened, as shown below:



\_\_\_ c. In the Business Integration view's tree, expand the EventEmitterTest → Mediation Logic → Flows; select EmitterTestFlow mediation flow, drag it over the assembly editor's canvas



\_\_\_\_d. In the Business Integration view's tree, expand the CustomerIDModule → Assembly Diagram; select Component1Export



- \_\_\_\_e. Drag it over the assembly editor's canvas
- \_\_\_\_\_f. Select **Import with SCA Binding** from the **Component Creation** window that pops-up upon dragging the **Component1Export**.

🚯 Component Creation	×
Select the type to create:	
Import with no Binding ☐ Import with SCA Binding	
OK Cance	<u>.</u>

#### \_\_\_ g. Click **OK**

\_\_\_\_h. The assembly diagram must look like the diagram shown below:



\_\_\_\_\_i. Right click the EmitterTestFlow component and select Generate Export → Web Service Binding from the context menu



\_\_\_\_j. Select soap/http from the Transport Selection pop-up window

🚯 Transport Selectio	n	×		
Select transport for CustomerDetailsExport1				
soap/http soap/jms				
1				
	ОК	Cancel		

\_\_\_ k. Click OK

\_\_\_\_I. The assembly diagram must look like the diagram shown below:



\_\_\_\_7. The Mediation Module and Mediation Flow creation is complete

### Part 3: Deploy the modules and configure Web services explorer

In this section of the lab, the Mediation Module is installed (or deployed) to the WebSphere ESB Server 6.0 test server and the Web services explorer is configured to prepare the ground for testing.

1. Start the **WebSphere ESB Server** if not started and **add modules** to the 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 ESB Integrated test environment, complete the steps below:

#### \_\_\_\_a. Open Servers View

\_\_\_ b. Select the WebSphere ESB Server v6.0 and right-click to select "( 🔍 ) Start" from the context menu

Properties Problems Console Console		\$	0 🖉 🖞 🔺 👘 🗗 🗖
Server	Host name	Status	State
WebSphere ESB Server v6.0	localhost	Stopped	Synchronized
4		8830)	

- \_\_\_\_ c. This takes some time. Wait for the server to start
- \_\_\_\_\_d. Add the **projects** to WebSphere ESB Server. In the servers view, right-click on WebSphere Process Server v6.0 and select "**Add and Remove Projects...**" from the context menu

	🏪 Add and remove projects	
	Run universal test client	
	Restart universal test client	
	Run administrative console	
	Reconnect debug process	
	📋 Create tables and data sources	
	Import server configuration from serve	er
Properties Problems 👫	Export server configuration to server	
Server	Launch	•
🛅 WebSphere ESB Servi	v6.0 iocainosc	
🛅 WebSphere Process S	rver v6.0 localhost	

**NOTE:** Note that the WebSphere 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 confi	gured on the server		
Move projects to the right to config	ure them on the server		
<u>A</u> vailable projects:		$\underline{C}$ onfigured projects:	
⊕( CustomerIDModuleApp EventEmitterTestApp	A <u>d</u> d > < <u>R</u> emove Add All >>		
	< <u>B</u> ack <u>N</u> ext >	Einish	Cancel

\_\_\_\_f. Click the Add-All>> button to move all projects to server

🚯 Add and Remove Projects	×
Add and Remove Projects Modify the projects that are configured on the set	rver
Move projects to the right to configure them on the	server
<u>A</u> vailable projects:	<u>C</u> onfigured projects:
Add :	EventEmitterTestApp
Add Al	>>
< <u>B</u> ack	Next > Einish Cancel

- \_\_\_ 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

Publishing CustomerIApp: (2%)	<b>III</b> Ó
-------------------------------	--------------

#### **Configure WebServices Explorer:**

- \_\_\_\_ 3. To Configure WebServices Explorer in the WebSphere Integration Developer, complete the steps below:
  - \_\_\_\_a. In the WebSphere Integration Developer's main menu, select Window > Preferences
  - \_\_\_\_b. The Preferences window opens

Ð-Workbench Ð- Ant	Workbench		
+ Ant			
Build Order	🗌 Always run in background		
E Business Integration	✓ Build automatically		
- Event Definition Editor	Refresh workspace automatically		
± Help 	Save automatically before build		
±Internet	Keep next/previous part dialog open		
Ð - Java Ð - LPEX Editor Ð - Modeling Ð - Rup/Debug	Workspace save interval (in minutes): 5		
± Server	Open mode		
Validation	Double click		
WebSphere Service Registry a	C Single click		
	Elect on hover		
	Copen when using arrow keys		
	Note: This preference may not take effect on all views		
	Restore Defaults Apply		

\_\_\_\_ c. From the **Preferences** window, expand **Workbench** and select **Capabilities**. Select the check box next to **Web Service Developer** in the Capabilities frame

Preferences	
Appearance     Capabilities     Colors and Fonts     Colors and Fonts     Compare/Patch     F Editors     File Associations     Keys     Label Decorations     Linked Resources     Local History     Perspectives     Search     Startup and Shutdown     Startup and Shutdown     Build Order     Business Integration     Fevent Definition Editor	Capabilities         Qapabilities:         Qapabilities:
Help     Install/Update     Install/Update	Description: Restore Defaults Apply
Import Export	OK Cancel

\_\_\_ d. Click OK

\_\_\_\_e. The WebServices Explorer is configured

#### Copy the EventEmitterTest Service.wsdl file into CustomerLibrary:

- 4. For the WebServices Explorer to resolve the business object type correctly, copy the CustomerDetailsExport1\_CustomerDetailsHttp\_Service.wsdl file into the CustomerLibrary project. To achieve this, complete the steps below:
  - \_\_\_\_a. Open the Java Perspective in the WebSphere Integration Developer. (Window  $\rightarrow$  Open Perspective  $\rightarrow$  Other.. and select Java from the Select Perspective window and click OK
  - \_\_\_\_ b. In the Java Perspective, expand the EventEmitterTest project and locate the CustomerDetailsExport1\_CustomerDetailsHttp\_Service.wsdl file. Right-click on this WSDL file and select Copy from the context menu



\_\_\_\_ c. Now, right-click the **CustomerLibrary** project and select **Paste** from the context menu

Ignore any warnings on the problems tab that pop up.



\_\_\_\_\_d. Now expand the **CustomerLibrary** project; right-click the

CustomerDetailsExport1\_CustomerDetailsHttp\_Service.wsdl file and select Web Services → Test with Web Services Explorer from the context menu

CustomerDetailsExport1_CustomerDetailsHttp_1 EventEmitterTest EventEmitterTestApp EventEmitterTestEJB EventEmitterTestEJBClient EventEmitterTestEJBClient EventEmitterTestWeb	Refactor          Import         Export         % Refresh	Alt+Shift+T >	-
	Run Validation Validate WSDL File Cenerate Documentation Run Debug Profile Tgam Compare With Replace With Link Utilities Web Services Source	) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) ) )	tion Resource th1 on node Messag EmitterTestFlor Test with Web Services Explorer Publish WSDL file
CustomerDetailsExport1_CustomerDetailsHttp_Serviv	P <u>r</u> operties	Alt+Enter	Generate Java bean skeleton Generate Client

\_\_\_\_e. The Web Services Explorer opens as shown below:

Web Services Explorer	
Sc. Navigator	Actions
ass vSDL Man ⊡-⊉ file:/C:/LabFiles602/WPS/EventEmitterPrimitive/workspace3/	2 WSDL Binding Details
CustomerDetailsExport1_CustomerDetailsHttpService     Log     Log	Shown below are the details for this <b>SDAP</b> <binding> element. Click parameters and invoke it or specify additional endpoints.</binding>
	- Operations
	Name Documentation
	getCustomerID
	Endpoints Add Remove
	Endpoints
	http://localhost:9080/EventEmitterTestWeb/sca/CustomerDetailsExport1
	Go Reset

\*\* Note that Status may show errors encountered while validating XML. Do not worry about them for now.

*Note*: If you are running against a server other than the localhost and or on non-default ports you will need to configure different endpoints for the Web services explorer to use. To configure the a new Endpoint, click the **Add** link next to **Endpoints**, and then the modify the endpoints URL to match your environment, select the check box next to the text filed and click on the **Go** button. Ensure that the Endpoints are successfully updated.

*	Endpoints (Add) Remove
	Endpoints
	http://localhost:9080/EventEmitterTestWeb/sca/CustomerDetailsExport1
	http://localhost:9081/EventEmitterTestWeb/sca/CustomerDetailsExport1
Go	Reset

## Part 4: Event generation test A and B using the Web services explorer

In this section of the lab two tests, test A and test B, are performed to produce events at the CEI Server. The Test A is to generate the request and response events with the value of the age element of the in the CustomerDetails business object; greater than 16 (>16). The Test B is to do the same with the value of the age less than 16 (<16).

#### **Event Generation Test A**

1. In the Web Services Explorer's Navigator pane, expand CustomerDetailsExport1\_CustomerDetailsHttpBinding and select the operation getCustomerID as shown below:

Web Services Explorer	
😵 Navigator 🛷 🖉	Actions
But WSDL Main     Def	Marke a WSDL Operation
E-2 The; CC; Labriesou2; WPS; EventEntitterPrint(VerPrin	Enter the parameters of this WSDL operation and click <b>Go</b> to invoke. Endpoints http://localhost:9080/EventEmitterTestWeb/sca/CustomerDetailsExport1
	<ul> <li>✓ getCustomerID</li> <li>✓ input1 □ nil?</li> </ul>
	<u>f_name</u> string <u>Add_Remove</u>
	T Values
	✓ Lname string Add Remove
	T Yalues
	✓ age int Add Remove
	T Values
	✓ address Add Remove
	Content
	Go Reset

- 2. Enter the following values in the following fields on the Actions pane of the explorer by clicking on the **Add** link to add a blank field for each of the values below:
  - \_\_\_a. f\_name: Fred
  - \_\_\_ b. I\_name: Smith
  - \_\_\_ c. age : 55

Web Services Explorer		
😪 Navigator 🔗	S C	
Solution     Solution	space3, 🔍 Invoke a WSDL Operation	
CostonerDetailsExport1_CostonerDetailsHttpBin	Inding Enter the parameters of this WSDL operation and click <b>Go</b> to invoke.	
	http://localhost:9080/EventEmitterTestWeb/sca/CustomerDetailsExport1	
	▼ <u>getCustomerID</u>	
	<ul> <li><u>input1</u> □ nil?</li> <li>f name string Add Remove</li> </ul>	
	□ Values	
	Fred	
	▼ Liname string Add Kemove Values	
	Smith	
	✓ age int Add Remove	
	address Add <u>Remove</u>	
	Go Reset	

- 3. Scroll down and click the Add link next address to add the address fields. Enter the following values for the address fields:
  - \_\_\_a. houseNumber: 1
  - \_\_\_b. street : High Street
  - \_\_\_\_\_c. postcode : S12JN

Web Services Explorer		
😪 Navigator 🔗 📿	Actions	
	55	
E-2 CustomerDetailsExport1_CustomerDetailsHttpService	address Add Remove	
CustomerDetailsExport1_CustomerDetailsHttpBinding     CustomerID		Content
	<u> → houseNumber</u> strin	ng Add Remove
		¥alues
	✓ <u>street</u> string <u>Add</u>	Remove
		Values
	High Street	
	✓ postcode string A	dd <u>Remove</u>
		Values
	S12JN	
	Go Reset	

- 4. Click on the **Go** button to generate the Events
- \_\_\_5. Ensure that there are no errors displayed in the Status pane at the bottom

i Status	Æ
	Source
<ul> <li>getCustomerIDResponse</li> </ul>	
customerID (string): ABCDE12345	

6. To confirm that the Events are generated, open the **Common Base Event browser** at <a href="http://hostname:<port>/ibm/console/cbebrowser/events/">http://hostname:<port>/ibm/console/cbebrowser/events/</a> as shown in the diagram below:

*Note:* The **hostname** is a fully qualified name of the machine and **port** is the **default\_http** port of the ESB server. Ex: <u>http://localhost:9080/ibm/console/cbebrowser/events/</u>

Also you can open the common base event browser by switching to the Business Integration perspective and in the server view, right-click on the server and select **launch->Common Base Event Browser** 

🗿 Common Base Even	t Browser - Microsoft Internet Explorer	_ 🗆 ×
<u>Eile E</u> dit <u>V</u> iew F <u>a</u> v	vorites Iools Help	200
Back Forward	💌 😰 🐔 🔎 💏 🧭 🙆 ד Stop Refresh Home Search Favorites History Mail	Print Edit
Address 🙆 http://localh	nost:9060/ibm/console/cbebrowser/events	💌 🛃 Go 🛛 Links 🂙
WebSphere soft Com	ware mon Base Event Browser	
Get Events   Help		
Event Views	Get Events	
All Events	Retrieve set of Common Base Events from the Common Events	Infrastructure to view.
BPEL Process Events	Event Data Store Properties	]
<u>User Data</u> <u>Events</u>	+Event Data Store	The JNDI name used to
Server Events	java:comp/env/eventsaccess	
Number of	*Event Group	The event group from which
events: O	All events	the events are retrieved
	Maximum Number of Events to Retrieve	The maximum number of
🛐 Done		Local intranet

7. From the Event Browser, on the top left of the navigation menus, click on the Get Events link and then the All Events link under Events View. Ensure that two events have been generated; one for EventEmitter1 on the request flow and the other for EventEmitter1 on the response flow as shown below:

🙆 Common Base Event Browser - Microsoft Internet Explorer	
Ele Edit Yew Pavorites Tools Help	
🔾 - 🕤 - 💽 👔 🔥 🔎 🔅 🥙 🙆 - 🤤 - Back Forward Stop Refresh Home Search Favorites History Mail Print Edit -	
Address 🙆 http://localhost:9060/ibm/console/cbebrowser/events	💌 🔁 Go 🛛 Links 🌺
WebSphere software Common Base Event Browser	IBM.
Get Events Help	
Event Views         All Events         BPEL Process    Select ^ Creation Time ^ Name ^ I	Priority ^ Severity ^ Failed
Events O 2006-11-12T19:03:31.656Z EventEmitterTest_EmitterTestFlow_EventEmitter1_Req	
Events O 2006-11-12T19:03:31.859Z EventEmitterTest_EmitterTestFlow_EventEmitter1_Res	J
Server Events	<u> </u>
Number of events: 2	
C Done	Local intranet

8. To view the contents of the Request Flow EventEmitter1, click on the Creation Time link or select the radio button next to EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Req The common base event browser will display the events in the Event Data pane as shown here:

E' Lis	Event Data List of all properties associated with the selected event.		
	Namo	Value	
	INdifie	Value	
	version	1.0.1	
	globalinstanceld	CE7CFF2326CFB5B58CA1DB72807D098D80	
	extensionName	EventEmitterTest_EmitterTestFlow_EventEmitter1_Req	
	localinstanceld		
	creationTime	2006-11-12T19:03:31.656Z	
	severity		
	msg		
	priority		
	sequenceNumber	9	
	repeatCount		
	elapsedTime		
	contextDataElement /WBISESSION_ID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163358211359;1007133850	
	contextDataElement / ECSCurrentID / contextValue	9.3.75.99;EventEmitterTest;null;;getCustomerID;1163358211359;1007133850	
	contextDataElement / ECSParentID / context∀alue	9.3.75.99;EventEmitterTest;;getCustomerID;1163358211359;1007133850	
	extendedDataElement / EventNature	CUSTOM	
	extendedDataElement / PayloadType	full	
- (	extendedDataElement / ModuleName	EventEmitterTest	
	extendedDataElement / MediationName	EventEmitter1	
_ (	extendedDataElement / Root	No Data	

- 9. Look for the following elements in the event to ensure the correct data has been generated:-
  - \_\_\_a. extensionName → EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Req
  - \_\_\_b. extendedDataElement/ModuleName → EventEmitterTest
  - \_\_\_ c. extendedDataElement/MediationName → EventEmitter1
  - \_\_\_\_d. extendedDataElement/Root → No Data
- 10. To view the contents of the Response Flow EventEmitter1, click on the Creation Time link or select the radio button next to EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Res. The common base event browser will display the events in the Event Data pane as shown below:

Event Data	Event Data		
List of all properties a	ist of all properties associated with the selected event.		
Nomo	Nome		
IName		value	
version		1.0.1	
globalinstanceld		CE7CFF2326CFB5B58CA1DB72807D261630	
extensionName		EventEmitterTest_EmitterTestFlow_EventEmitter1_Res	
localinstanceid			
creationTime		2006-11-12T19:03:31.859Z	
severity			
msg			
priority			
sequenceNumber		10	
repeatCount			
elapsedTime			
contextDataElement //	WBISESSION_ID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163358211359;1007133850	
contextDataElement /	ECSCurrentID / contextValue	9.3.75.99;EventEmitterTest;null;;getCustomerID;1163358211359;1007133850	
contextDataElement /	ECSParentID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163358211359;1007133850	
extendedDataElement	/EventNature	CUSTOM	
extendedDataElement	/PayloadType	full	
extendedDataElement	/ ModuleName	EventEmitterTest	
extendedDataElement	/ MediationName	EventEmitter1	
extendedDataElement	/ Root	/body/getCustomerIDResponse/customerID	
extendedDataElement	/Message	ABCDE12345	

\_\_\_\_ 11. Look for the following elements in the event to ensure the correct data has been generated:-

- \_\_\_\_a. extensionName → EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Res
- \_\_\_b. extendedDataElement/ModuleName → EventEmitterTest
- \_\_\_ c. extendedDataElement/MediationName → EventEmitter1
- \_\_\_\_d. extendedDataElement/Root → /body/getCustomerIDResponse/customerID
- \_\_\_\_e. extendedDataElement/Message → ABCD12345
- \_\_\_ 12. The Event Generation Test A is complete

#### **Event Generation Test B**

13. In the Web Services Explorer's Navigator pane, expand CustomerDetailsExport1\_CustomerDetailsHttpBinding and select the operation getCustomerID and enter the following values in their respective fields:

(Note that you may need to delete the old values from their fields and replace them with these new ones)

\_\_\_a.f\_name : Harry

\_\_\_\_b. I\_\_name : Brown

\_\_\_\_ c. age : **13** 

#### Address

- \_\_\_\_d. houseNumber : 55
- \_\_\_\_e. street : Long Lane
- \_\_\_\_f. postcode : ABC23
- \_\_\_\_\_ 14. The Explorer with the entered values must look as shown below:

Web Services Explorer		
😵 Navigator	Actions	
品 WSDL Main	Harry	
⊡ A file:/C:/LabFiles602/WPS/EventEmitterPrimitive/workspace ⊡ A CustomerDetailsExport1 CustomerDetailsHttpService	✓ Lname string Add Remove	
O CustomerDetailsExport1_CustomerDetailsHttpBindin	Values	
	Brown	
	✓ age int Add Remove	
	T Yalues	
	✓ address Add Remove	
	Content	
	houseNumber string Add Remove	
	T Values	
	55	
	✓ street string Add Remove	
	T Values	
	Long Lane	
	postcode string Add Remove	
	Values	
	ABC23	

- \_\_\_\_\_15. Click on the Go button to generate the Events
- 16. Follow the steps 11 and 12 of this section to view the events generated. There must be two more events added to the number of events listed as result of Test A in the common base event browser; one for EventEmitter2 on the request flow and the other for EventEmitter1 on the response flow as shown here:

🚰 Common Base Event	Browser - Microsoft Internet Explorer	
<u>File Edit View Favo</u>	orites Iools Help	
Back Forward	💌 🖻 🙆 🔎 🔅 🥙 🎧 🖶 🚽 Stop Refresh Home Search Favorites History Mail Print Edit	
Address 🥘 http://localho	st:9060/ibm/console/cbebrowser/events	💌 🛃 Go 🛛 Links
WebSphere softw Comm	are non Base Event Browser	
Get Events   Help		
Event Views	👾 🧐 🖉 📄 📅 😰 💷 Select Action 🔽 Go	
BPEL Process	Select ^ Creation Time ^ Name ^ Priority ^	Severity ^ Failed
Events	O 2006-11-12T19:03:31.656Z EventEmitterTest_EmitterTestFlow_EventEmitter1_Req	
User Data Fuents	O 2006-11-12T19:03:31.859Z EventEmitterTest_EmitterTestFlow_EventEmitter1_Res	
Server Events	C 2006-11-12T19:49:52.016Z JuniorAccount	
	O 2006-11-12T19:49:52:109Z EventEmitterTest_EmitterTestFlow_EventEmitter1_Res	
Number of		×
Done		S Local intranet

17. To view the contents of the Request Flow EventEmitter2, click on the Creation Time link or select the radio button next to JuniorAccount. The common base event browser will display the events in the Event Data pane as shown below:

extensionName	JuniorAccount
localinstanceld	
creationTime	2006-11-12T19:49:52.016Z
severity	
msg	
priority	
sequenceNumber	11
repeatCount	
elapsedTime	
contextDataElement /WBISESSION_ID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163360992000;1252975837
contextDataElement / ECSCurrentID / contextValue	9.3.75.99;EventEmitterTest;null;;getCustomerID;1163360992000;1252975837
contextDataElement / ECSParentID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163360992000;1252975837
extendedDataElement / EventNature	CUSTOM
extendedDataElement / PayloadType	full
extendedDataElement / ModuleName	EventEmitterTest
extendedDataElement / MediationName	EventEmitter2
extendedDataElement / Root	/body
extendedDataElement / Message	
extendedDataElement / Message / getCustomerID	
extendedDataElement / Message / getCustomerID / input1	
extendedDataElement / Message / getCustomerID / input1 / f_name	Harry
extendedDataElement / Message / getCustomerID / input1 / I_name	Brown
extendedDataElement / Message / getCustomerID / input1 / age	13
extendedDataElement / Message / getCustomerID / input1 / address	
extendedDataElement / Message / getCustomerID / input1 / address / houseNumber	55
extendedDataElement / Message / getCustomerID / input1 / address / street	Long Lane

\_ 18. Look for the following elements in the event to ensure the correct data has been generated:-

- $\_$  a. extensionName  $\rightarrow$  JuniorAccount
- \_\_\_b. extendedDataElement/ModuleName → EventEmitterTest
- \_\_\_ c. extendedDataElement/MediationName → EventEmitter2
- \_\_\_d. extendedDataElement/Root → /body
- \_\_\_\_e. extendedDataElement / Message / getCustomerID / input1 / f\_name : Harry
- \_\_\_\_f. extendedDataElement / Message / getCustomerID / input1 / I\_name : Brown
- \_\_\_\_g. extendedDataElement / Message / getCustomerID / input1 / age : 13
- \_\_\_h. extendedDataElement / Message / getCustomerID / input1 / address / houseNumber : 55
- \_\_\_\_i. extendedDataElement / Message / getCustomerID / input1 / address / street : Long Lane
- \_\_\_\_j. extendedDataElement / Message / getCustomerID / input1 / address / postcode : ABC23
- 19. To view the contents of the Response Flow **EventEmitter1**, click on the **latest** Creation Time link or select the radio button next to **EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Res**. The common base event browser will display the events in the Event Data pane as shown below:

#### Event Data

List of all properties associated with the selected event.

Name	Value
version	1.0.1
globalinstanceld	CE7CFF2326CFB5B58CA1DB7286F64E91D0
extensionName	EventEmitterTest_EmitterTestFlow_EventEmitter1_Res
localinstanceld	
creationTime	2006-11-12T19:49:52.109Z
severity	
msg	
priority	
sequenceNumber	12
repeatCount	
elapsedTime	
contextDataElement /WBISESSION_ID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163360992000;1252975837
contextDataElement / ECSCurrentID / contextValue	9.3.75.99;EventEmitterTest;null;;getCustomerID;1163360992000;1252975837
contextDataElement / ECSParentID / contextValue	9.3.75.99;EventEmitterTest;;getCustomerID;1163360992000;1252975837
extendedDataElement / EventNature	CUSTOM
extendedDataElement / PayloadType	full
extendedDataElement / ModuleName	EventEmitterTest
extendedDataElement / MediationName	EventEmitter1
extendedDataElement / Root	/body/getCustomerIDResponse/customerID
extendedDataElement / Message	ABCDE12345

20. Look for the following elements in the event to ensure the correct data has been generated:-

- \_\_\_a. extensionName → EventEmitterTest\_EmitterTestFlow\_EventEmitter1\_Res
- \_\_\_b. extendedDataElement/ModuleName → EventEmitterTest
- \_\_\_c. extendedDataElement/MediationName → EventEmitter1
- \_\_\_\_d. extendedDataElement/Root → /body/getCustomerIDResponse/customerID
- \_\_\_\_e. extendedDataElement/Message  $\rightarrow$  ABCDE12345
- \_\_\_\_\_ 21. The Event Generation Test B is complete
- \_\_\_\_\_ 22. Close the Web Services Explorer

### Part 5: Event generation test C using the test client

In this section of the lab, a Test C is performed to produce events at the CEI Server using a Test Client. Note that the Test A and Test B which were done using a WebServices Explorer are combined to perform a Test C using a Test Client rather than a WebServices Explorer.

*Note*: To avoid confusion, delete all the events generated as result of the actions during the course of Part 4. To achieve this:

 $\rightarrow$  Open a command line window and change directory to <WID\_HOME>\pf\esb\bin> that is, E:\WID602\pf\esb\bin

*Note:* If you are testing on a remote server, go to the <WAS\_HOME>bin directory of the server and issue the command as shown (possibly substituting .sh for the file extension) and substituting <WAS\_HOME>/profiles/<PROFILE\_NAME>event/bin/eventpurge.jacl for the jacl script name.

 $\rightarrow$ Run the following JACL script to delete all the events:

wsadmin.bat -f E:\WID602\pf\esb\event\bin\eventpurge.jacl -group "All events" -seconds 0

🔤 C:\WINDOWS\system32\cmd.exe

C:\WID602\pf\esb\bin>wsadmin.bat -f C:\WID602\pf\esb\event\bin\eventpurge.jacl group "All events" -seconds 0 WASX7209I: Connected to process "server1" on node esbNode using SOAP connector; The type of process is: UnManagedProcess WASX7303I: The following options are passed to the scripting environment and are available as argument that is stored in the argv variable: "[-group, All events , -seconds, 0]" 6 Event(s) deleted. C:\WID602\pf\esb\bin>\_

- 1. On WebSphere Integration Developer, switch to the Business Integration Perspective. To switch to this perspective: Window → Open Perspective → Other... and select Business Integration from the Select Perspective window and click OK
- 2. In the Business Integration view's tree, expand the EventEmitterTest mediation module and double-click the mediation module assembly (<sup>1</sup> Assembly Diagram) to open it with the assembly editor
- Right-click anywhere on the white space on the canvas and select **Test Module** from the context menu
- 4. The Test client for **EventEmitterTest** mediation module is opened

🕄 Assembly Diagram: EventEmitterTest 🛛 🗏 *EventEmitterTest _Test 🗙					
Events				ÅÞ Å	¥ 🂫 🖽
Select the component, interface, and operation you would like to invoke. Click Continue to run.					
Events	General Pr	operties			
	▼ Detailed Pr	roperties			
	Configuration:	Default Module	Test		
	Module:	EventEmitterTe	st		
	Component:	CustomerDetail	sExport1		
	Interface:	CustomerDetail	5		
	Operation:	getCustomerID			
	Initial reguest p	parameters			
	Name		Туре	Value	
	🖃 input 1		CustomerDetails		
	f_name		string		
	I_name		string		
	age		Int Adduces	U	
	- address	Alumbar	daress		
	stree	enander st	string		
	posto	code	string		
Gurah Gardinmational	Data Pool				Conti

#### 5. Enter the same values in the fields, used for **Event Generation Test A** of Part4, as shown below:

🕄 Assembly Diagram: EventEmitterTest 🛛 🖹 *EventEmitterTest_Test	×			₽
Events			å 🖧 🖪	
Select the component, interface, and operation you would like to in	voke, Click Contini	ue to run.		
Events	General Prop	perties		
······∲▶ Invoke	<ul> <li>Detailed Prop</li> </ul>	perties		_
	Configuration:	Default Module Test		•
	Module: E	EventEmitterTest		•
	Component:	CustomerDetailsExport1		•
	Interface:	CustomerDetails		
	Operation:	getCustomerID		•
	Initial reguest par	rameters		
	Name	Туре	Value	
	🖃 input 1	CustomerDetails		
	f_name	string	Fred	
	I_name	string	Smith	
	age	int	55	_
	- address	Address		
	street	iu string	I High Street	
	postcoc	le string	S121N	<b>.</b>
	postcoc	so soning		
	Data Pool		Contir	
Events Configurations				<u> </u>

\_\_\_a. Click Continue

\_\_\_\_ b. Select WebSphere ESB Server 6.0 as the deployment location from the pop-up window when prompted

*Note*: By default WebSphere Process Server 6.0 is selected as the deployment location. Ensure the WebSphere ESB Server 6.0 is selected as a deployment location.

n Deployment Location		×
Select Deployment Location This server instance is currently running.		E
Deployment location:		
WebSphere Process Servers     WebSphere Process Server v6.0     WebSphere ESB Server v6.0     WebSphere ESB Server v6.0		New <u>S</u> erver
	Einis	sh Cancel

- \_\_\_ c. Click Finish
- \_\_\_\_\_6. Ensure the following is resulted on a successful execution:

😮 Assembly Diagram: EventEmitterTest 🛛 📄 *EventEmitterTest_Test	×			
Events	General Properti	es		
□ Invoke (CustomerDetailsExport1:getCustomerID) □ Invoke (CustomerDetailsExport1:getCustomerID) □ Invoke (CustomerDetailsExport1:getCustomerID) □ Request (CustomerDetailsExport1> EmitterTestF □ Invoke (CustomerDetailsExport1> EmitterTestF	Detailed Propert     Module: EventE     Component: Custon     Interface: Custon	ies <u>imitterTest</u> nerDetailsExport1 perDetails		
Response (EmitterTestFlow < Import1:getCuston     Response (CustomerDetailsExport1 < EmitterTes     Return (CustomerDetailsExport1:getCustomerID)     Stopped	Operation: getCus Return parameters:	tomerID		
	Name	Туре	Value	
		String	ABCDE12345	
Events Configurations				-

- 7. Now follow the Steps 11 to 17 of Part4 to view the events generated and the test is successful
- 8. Close the Test Client and say no to save changes when prompted
- 9. Run the test again and enter the same values in the fields, used for **Event Generation Test B** of Part4
  - \_\_\_\_a. In the Business Integration view's tree, expand the **EventEmitterTest** mediation module and double-click the mediation module assembly (<sup>-\_\_\_\_</sup> Assembly Diagram) to open it with the assembly editor
  - \_\_\_\_b. Right-click any where on the canvas and select Test Module from the context menu
  - \_\_\_\_ c. Enter the same values in the fields as used for Event Generation Test B of Part4, as shown below:

<ul> <li>▶ Invoke</li> <li>▶ Detailed Properties</li> <li>Configuration: Default Module Test</li> <li>Module:</li> <li>EventEmitterTest</li> <li>Component:</li> <li>CustomerDetailsExport1</li> <li>Interface:</li> <li>CustomerDetails</li> <li>Operation:</li> <li>getCustomerID</li> <li>Initial reguest parameters</li> <li>Name</li> <li>Type</li> <li>Value</li> <li>input1</li> <li>CustomerDetails</li> <li>f_name</li> <li>string</li> <li>Brown</li> <li>age</li> <li>int</li> <li>13</li> <li>address</li> <li>Address</li> <li>houseNu</li> <li>string</li> <li>55</li> <li>street</li> <li>string</li> <li>Long Lane</li> <li>postcode</li> <li>string</li> <li>ABC23</li> </ul>	rents	General Properties	
Configuration:       Default Module Test         Module:       EventEmitterTest         Component:       CustomerDetailsExport1         Interface:       CustomerDetails         Operation:       getCustomerID         Initial reguest parameters       Yalue         Initial reguest parameters       Yalue         Input1       CustomerDetails         f_name       string       Harry         I_name       string       Brown         age       int       13         address       Address       Address         houseNu       string       55         street       string       Long Lane         postcode       string       ABC23       V	∲► Invoke	<ul> <li>Detailed Properties</li> </ul>	
Module:       EventEmitterTest         Component:       CustomerDetailsExport1         Interface:       CustomerDetails         Operation:       getCustomerID         Initial reguest parameters         Name       Type         Value         input1       CustomerDetails         f_name       string         age       int         int       13         address       Address         houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		Configuration: Default Module Test	•
Component:       CustomerDetailsExport1         Interface:       CustomerDetails         Operation:       getCustomerID         Initial reguest parameters         Name       Type       Value         input1       CustomerDetails       Finame         f_name       string       Harry         age       int       13         age       int       13         address       Address       S5         street       string       Long Lane         postcode       string       ABC23       Value		Module: EventEmitterTest	•
Interface:       CustomerDetails         Operation:       getCustomerID         Initial reguest parameters         Name       Type       Value         □ input1       CustomerDetails          f_name       string       Harry         l_name       string       Brown         age       int       13         □ address       Address          houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		Component: CustomerDetailsExport1	•
Operation:       getCustomerID         Initial reguest parameters         Name       Type       Value         input1       CustomerDetails         f_name       string       Harry         l_name       string       Brown         age       int       13         address       Address       Address         houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		Interface: CustomerDetails	•
Name       Type       Value         input1       CustomerDetails       Image         f_name       string       Harry         l_name       string       Brown         age       int       13         maddress       Address       String         houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		Operation: getCustomerID	•
NameTypeValueinput1CustomerDetailsf_namestringl_namestringageintindressAddressAddressAddresshouseNustringstreetstringLong LanepostcodestringABC23		Initial reguest parameters	
□ input1       CustomerDetails         f_name       string       Harry         l_name       string       Brown         age       int       13         □ address       Address          houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		Name Type Va	alue
f_name       string       Harry         l_name       string       Brown         age       int       13         address       Address          houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		🖃 input 1 CustomerDetails	
I_namestringBrownageint13addressAddress-houseNustring55streetstringLong Lanepostcodestring-		f_name string Ha	arry
age     int     13       □ address     Address     Address       houseNu     string     55       street     string     Long Lane       postcode     string     ABC23		I_name string Br	rown
□ address       Address         houseNu       string       55         street       string       Long Lane         postcode       string       ABC23		age int 13	3
houseNu     string     55       street     string     Long Lane       postcode     string     ABC23		address Address	
street string Long Lane postcode string ABC23		houseNu string 55	5
postcode string ABC23		street string Lo	ong Lane
		postcode string AE	3C23 💽

- \_\_\_\_ d. Click Continue
- \_\_\_\_e. Select WebSphere ESB Server 6.0 as the deployment location from the pop-up window when prompted
- \_\_\_ f. Click Finish
- \_\_\_\_g. Ensure the following is resulted on a successful execution:

🕄 Assembly Diagram: EventEmitterTest 🔰 🖹 *EventEmitterTest_Tes	X			
Events	General Properti	es		
Invoke (CustomerDetailsExport1:getCustomerID)  Started  Started  Request (CustomerDetailsExport1:getCustomerID)  Request (CustomerDetailsExport1> EmitterTestF  Request (EmitterTestFlow> Import1:getCustome  Response (EmitterTestFlow < Import1:getCuston  Response (CustomerDetailsExport1 < EmitterTest  Response (CustomerDetailsExport1 < EmitterTest	Detailed Propert     Module: EventE     Component: Custom     Interface: Custom     Operation: getCus	ies i <u>mitterTest</u> herDetailsExport1 herDetails :tomerID		
Stopped	Return parameters:	Tupo	Value	
	customerID	String	ABCDE12345	
Events Configurations				-

- \_\_\_\_\_ 10. Now follow the Steps 18 to 25 of Part4 to view the events generated and the test is successful
- \_\_\_\_\_ 11. Close the Test Client and say no to save changes when prompted
- \_\_\_\_\_ 12. The Event Generation Test C using the Test Client is complete

### Part 6: Save the work and clean up server

- \_\_\_\_\_1. 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:
    - CustomerIDModule
    - CustomerLibrary
    - EventEmitterTest
  - \_\_\_\_d. Save in C:/LabFiles602/WESB/EventEmitterPrimitive/
  - \_\_\_\_e. Name the project interchange WESB\_EventEmitterPrimitive\_Solution\_Pl.zip
  - 2. Remove all the projects and **clean** up the ESB Server.
    - \_\_\_\_a. Right-click on WebSphere ESB Server v6.0 (once started) and select Add and Remove Projects...
    - \_\_\_\_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 created a Mediation Module and a Mediation Flow. You created the Request and Response Flows for the Mediation Flow. Further you visually composed the Mediation Module in the Assembly Editor. Finally you tested the Mediation Module using a Web Services Explorer and a Test Client.

## **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/EventEmitterPromitive/ WESB\_EventEmitterPrimitive\_Solution\_PI.zip
  - \_\_\_\_ c. Click Finish button
- 2. Continue with Part 3, Part 4 & Part 5 of this LAB

# 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]	9
WebSphere profile name:	<b>V</b>
Server connection type and admin port     Server connection type and admin port     Server connection type and admin port     Server connection type and admin port	
ORB bootstrap port: 9131	
O 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	
<ul> <li>BASE, Express or unmanaged Network Deployment server</li> </ul>	
Network Deployment server	
Network Deployment server name:	
The server name is in the form of: <cell name="">/<node name="">/<server name=""> For example, localhost/localhost/server1.</server></node></cell>	
Detect Click this button to detect the server type.	
< Back Next > Finish	Cancel

- \_\_\_ g. Click Finish
- \_\_\_h. The new server should be seen in the Server view
- 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>'

userid: <USERID>

#### pwd: **<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

This page is left intentionally blank.