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

WebSphere Enterprise Service Bus lab 1 – Add mediation, mediation flow and message logger

What this exercise is about

The objective of this lab is to provide you with an initial understanding of how to create a mediation module and mediation flow in WebSphere Integration Developer V6.0.2. Then you will run a mediation module with Message Logger mediation on the WebSphere Enterprise Service Bus V6.0 server. You will also learn how to check message logger results in a Cloudscape database using the cview GUI that comes with Cloudscape.

Lab requirements

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

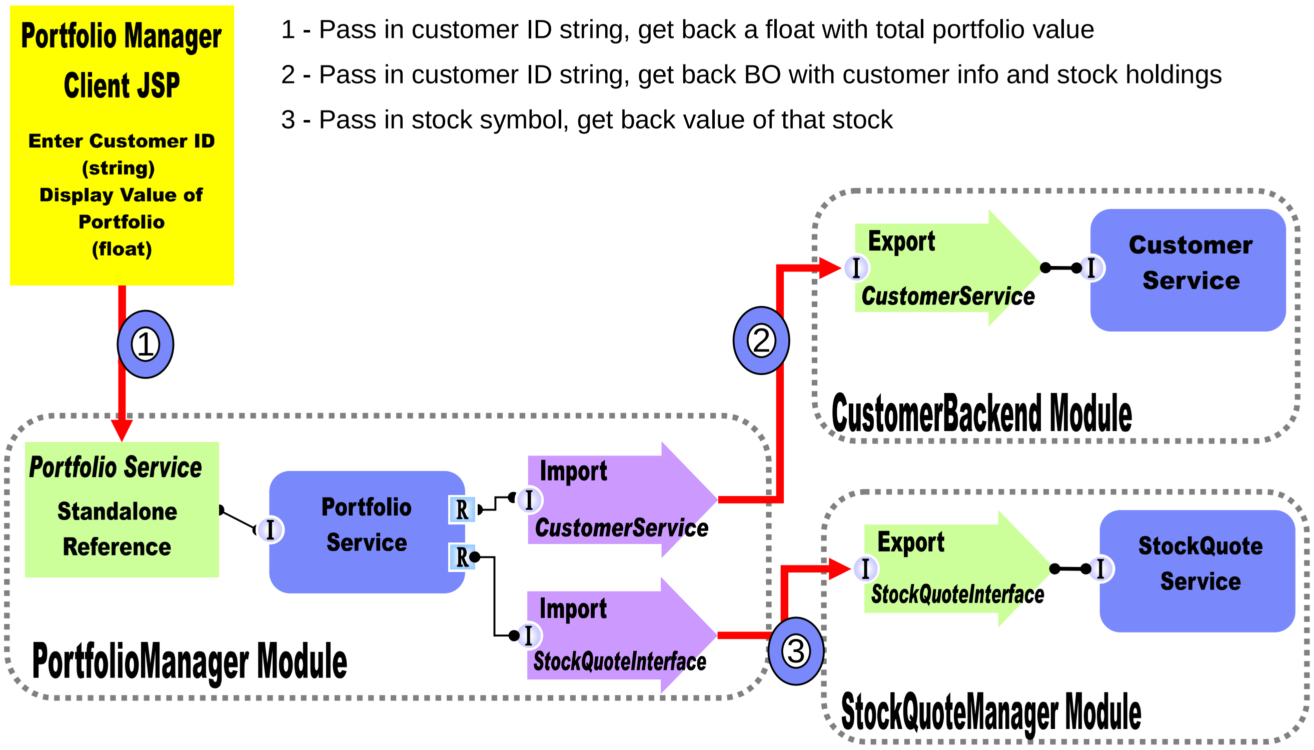
At the end of this lab you should be able to:

- Import project interchange files into the WebSphere Integration Developer V6.0.2 development environment.
- Create and edit mediation modules and mediation flows.
- Generate implementation and binding from the development environment.
- Navigate the Properties View for mediation information.
- Create and learn about a message logger primitive and test by running a JSP on the WebSphere Enterprise Service Bus V6.0 server.
- View Cloudscape database information using the Cview GUI.

Introduction

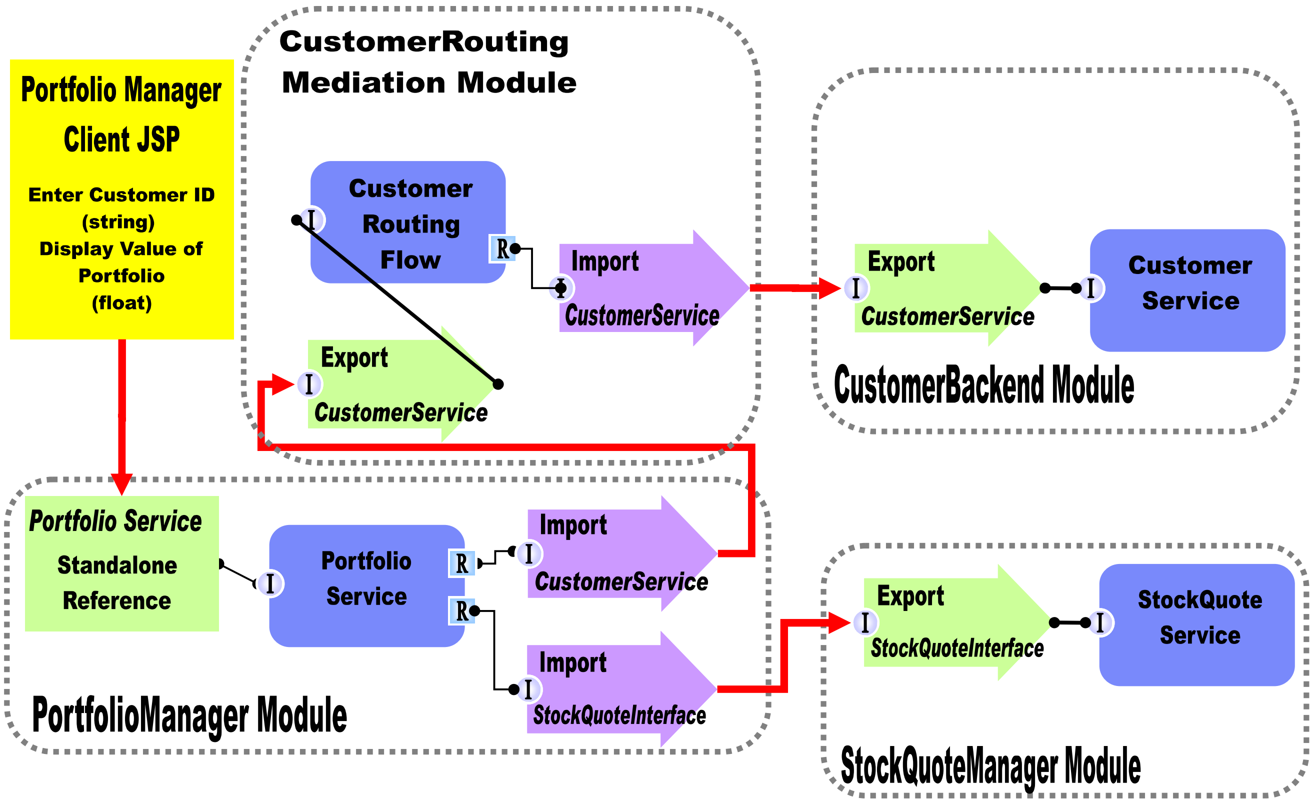
In this lab, you will start by importing a project interchange file that has sample SCA application which will provide the mediation module you will create a starting point. You will be creating a brand new mediation module and learning about what makes the mediation module different from any of the other business integration modules. You will also create a simple mediation flow with only one type of mediation, message logger mediation, to start off with. The job of the message logger mediation is to send data, whether the /body, /context, or /header, to a cloudscape database where you will be able to see what data was sent to the database. The rest of the mediations will be added in Lab 2 of this lab series, and all will be debugged in Lab 3 to show you how the mediations work. In Lab 4, you will work with the administrative console to make changes to live applications, instead of changing information from the development environment.

The sample application is a stock checking SCA application. The application has a JSP that allows a user to enter a customer ID which is fed into a call to the portfolio service by way of a stand-alone reference. From here the portfolio service will first call the customer service to get a customer's account information like what stocks they own and how many shares of that stock. Portfolio service will then call the StockQuote service to see the value of those stocks. Finally, the portfolio service returns a sum to the JSP. Here is a diagram of the project you are importing in this lab.

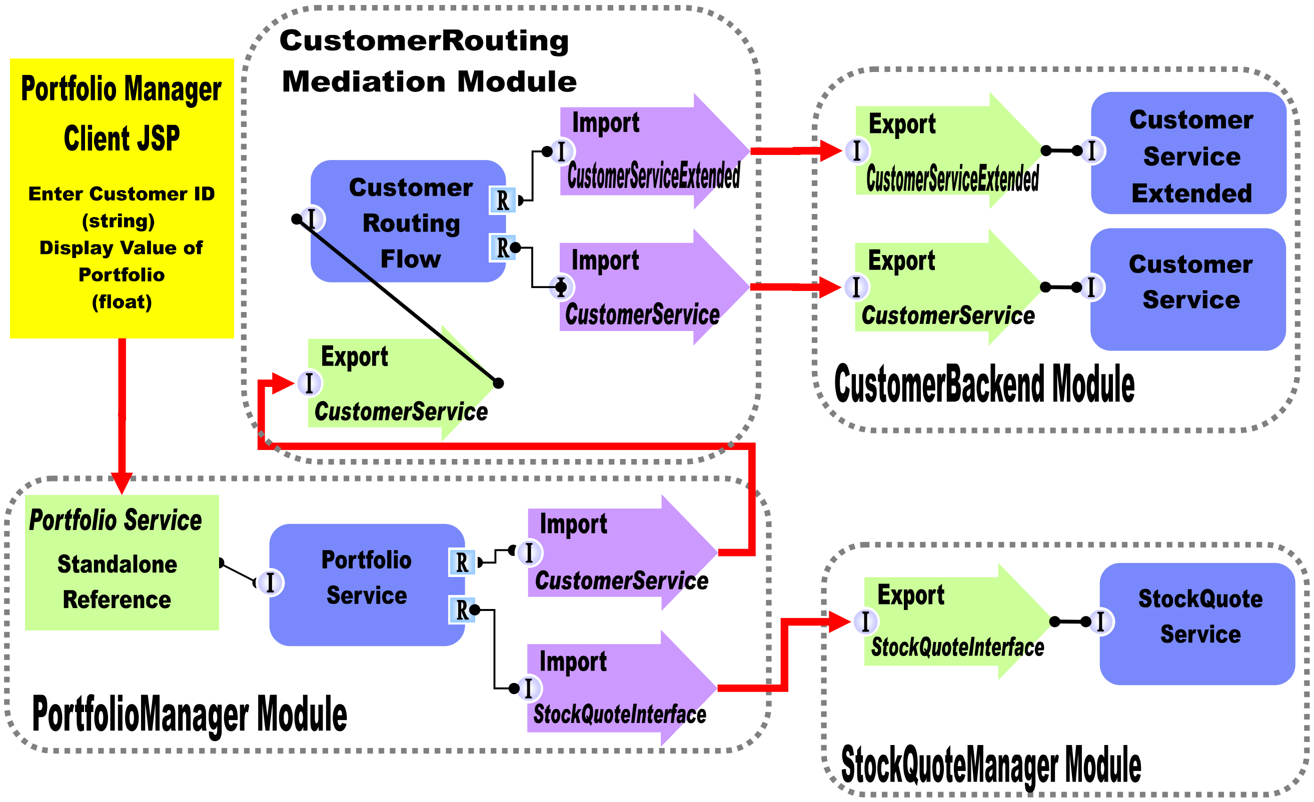

Portfolio Manager Client JSP
Enter Customer ID (string)
Display Value of Portfolio (float)
1 - Pass in customer ID string, get back a float with total portfolio value
2 - Pass in customer ID string, get back BO with customer info and stock holdings
3 - Pass in stock symbol, get back value of that stock
1
2
3
Export
CustomerService
Customer Service
CustomerBackend Module
Portfolio Service
Standalone Reference
Portfolio Service
R
R
I
Import
CustomerService
Import
StockQuoteInterface
PortfolioManager Module
Export
StockQuoteInterface
StockQuote Service
StockQuoteManager Module

This original application is great, but the company that is running this application is going to need to distribute customer data across new multiple "CustomerBackend Modules and easily enhance end-user functionality.

In Lab 1 of this lab series, you will change the application from the above diagram to the one below. You will add a **CustomerRoutingMediation** Module between the **PortfolioManager** service and the CustomerBackend service to allow you access to message information being passed between them. You will create a mediation flow called **CustomerRoutingFlow** that will use a Message Logger mediation to store message information in a cloudscape database.



In Lab2, you will add to the Message Logger mediation primitive with a couple others available in WebSphere Integration Developer V6.0.2. The main goal for Lab2 will be to create another backend, **CustomerServiceExtended**, and use mediations to help decide which backend to use. You will first add a Custom Mediation. The Custom Mediation is a mediation primitive that is used when you want some custom functionality from mediation. For this custom mediation, it will use a Java snippet that will extract a two digit prefix from the customer ID and place 2 digit it in the transient context (a business object). Think of transient context as a scratchpad, a place for information to be stored in a message as it passes through a flow. You will then add a Database Lookup primitive that uses that two digit prefix from transient context as key to lookup a backend identifier to place into transient context. The Customer ID prefixes with 11, 22, 33, and 44 will go to the CustomerService backend. Customer ID prefixes with 55, 66, 77, 88, and 99 will go to a **CustomerServiceExtended** backend. To determine the message routing based on backend identifier, you will add a Message Filter primitive. The old backend will go directly to the callout for CustomerService and the new backend will go to the XSL Transformation. The XSL Transformation will transform the CustomerService business object (BO) into the **CustomerServiceExtended** business object, and then pass the newly formed message to the callout for **CustomerServiceExtendedPartner** instead of the **CustomerServicePartner**. Notice that this is just the request. There is also a flow for the response. On the response side, the **CustomerServicePartner** callout response will go directly to the CustomerService input response. However, the **CustomerServiceExtended** callout response will need to go back through XSL Transformation mediation in order to transform the response body from the **CustomerServiceExtended** business object back to the CustomerService business object. Feeding the **CustomerServiceExtended** business object to application would end in error because the application only knows how to work with the CustomerService business object. This is the reason why you need an XSL Transformation mediation primitive; to map one business object to another. The diagram will now look like the one below.


CustomerRouting Mediation Module
Portfolio Manager Client JSP
Enter Customer ID (string)
Display Value of Portfolio (float)
Customer Routing Flow
Import CustomerServiceExtended
Import CustomerService
Export CustomerService
Export CustomerServiceExtended
Customer Service Extended
Export CustomerService
Customer Service
CustomerBackend Module
Portfolio Service Standalone Reference
Portfolio Service
Import CustomerService
Import StockQuoteInterface
PortfolioManager Module
Export StockQuoteInterface
StockQuote Service
StockQuoteManager Module

In Lab 3 of this lab series, you will use the Visual Debugger to step through the application in order to see what is happening behind the scenes.

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
<LAB_NAME>	Lab1	Lab1
<WID_HOME>	C:\Program Files\IBM\ID\6.0	/opt/IBM/WebSphere/ID/6.0
<LAB_FILES>	C:\Labfiles602	/tmp/Labfiles602
<WORKSPACE>	C:\Labfiles602\WESB\Lab1\workspace	/tmp/Labfiles602/WESB/Lab1/workspace
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles602\WESB\Lab1\solution	/tmp/Labfiles602/WESB/Lab1/solution

Windows user 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. The following table is related to where you are running remote test environment:

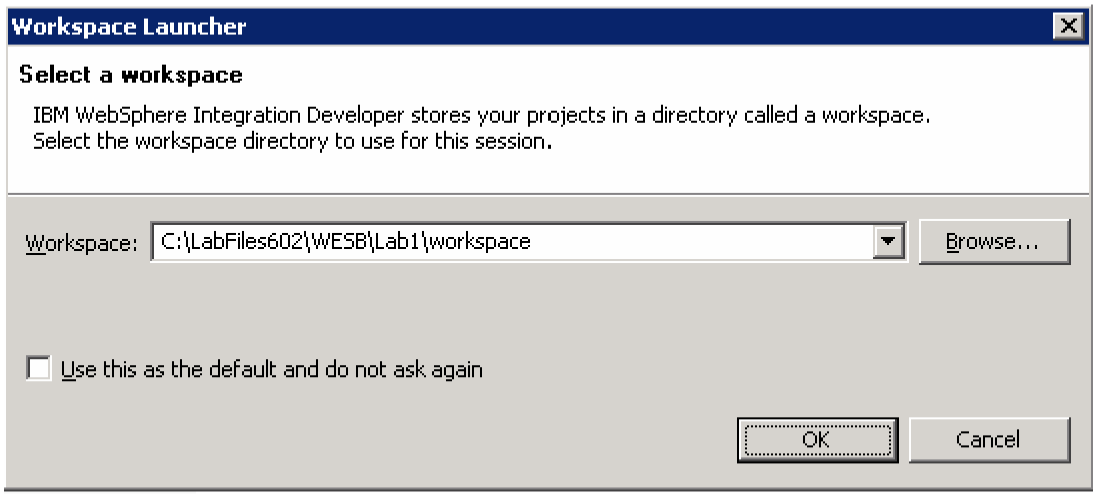
Reference Variable	Example: Remote Windows test server location	Example: Remote z/OS® test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	cl1sr01	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/cl1cell/AppServerNode1	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<BOOTSTRAP_PORT>	2809	2809	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	cl1admin	
<PASSWORD>	N/A	fr1day	

Instructions for using a remote testing environment, such as z/OS, AIX or Solaris, can be found at the end of this document, in the section "**Task: Adding Remote Server to WebSphere Integration Developer Test Environment**".

Part 1: Prepare environment for lab

In this section of the lab, you will be importing all projects inside the WPIv602_ESB_StartLab1_PI.zip project interchange file into your workspace. Remember this is the sample SCA application that you are going to add ESB specific mediations to.

____ 1. Start WebSphere Integration Developer V6.0.2 with a workspace location of **C:\LabFiles602\WESB\Lab1\workspace**.



Workspace Launcher

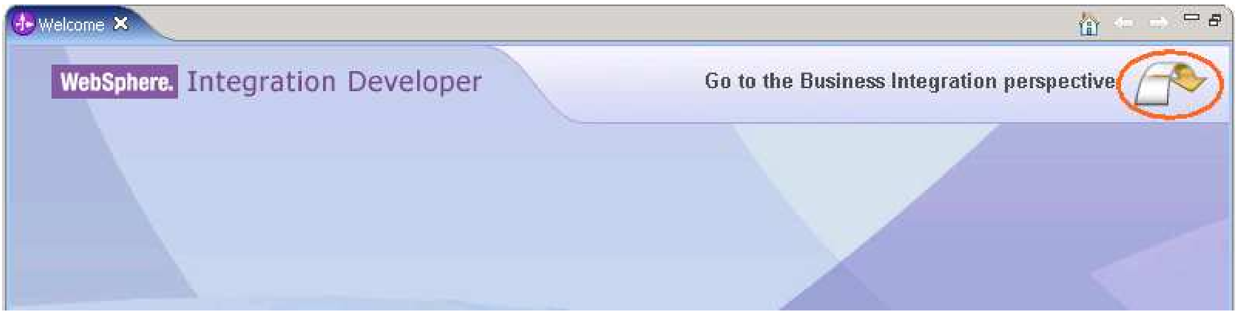
Select a workspace

IBM WebSphere Integration Developer stores your projects in a directory called a workspace.
Select the workspace directory to use for this session.

Workspace: C:\LabFiles602\WESB\Lab1\workspace

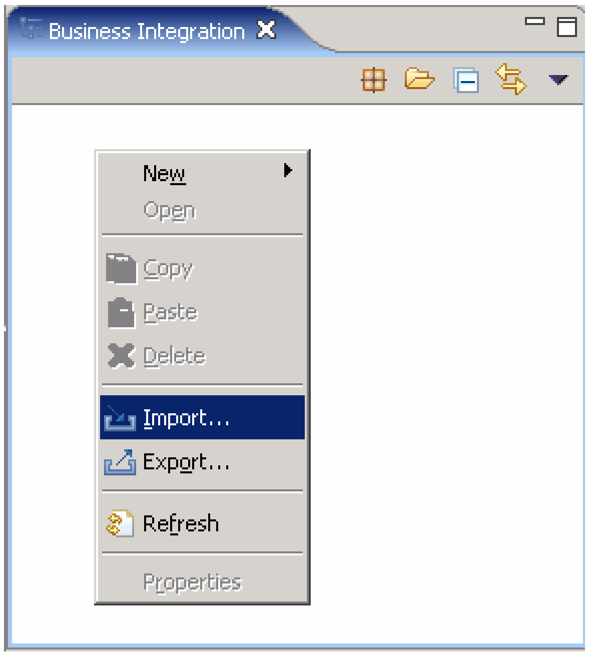
☐ Use this as the default and do not ask again

____ 2. Click the curved arrow at top right to **go to Business Integration perspective**.


Welcome
WebSphere. Integration Developer
Go to the Business Integration perspective

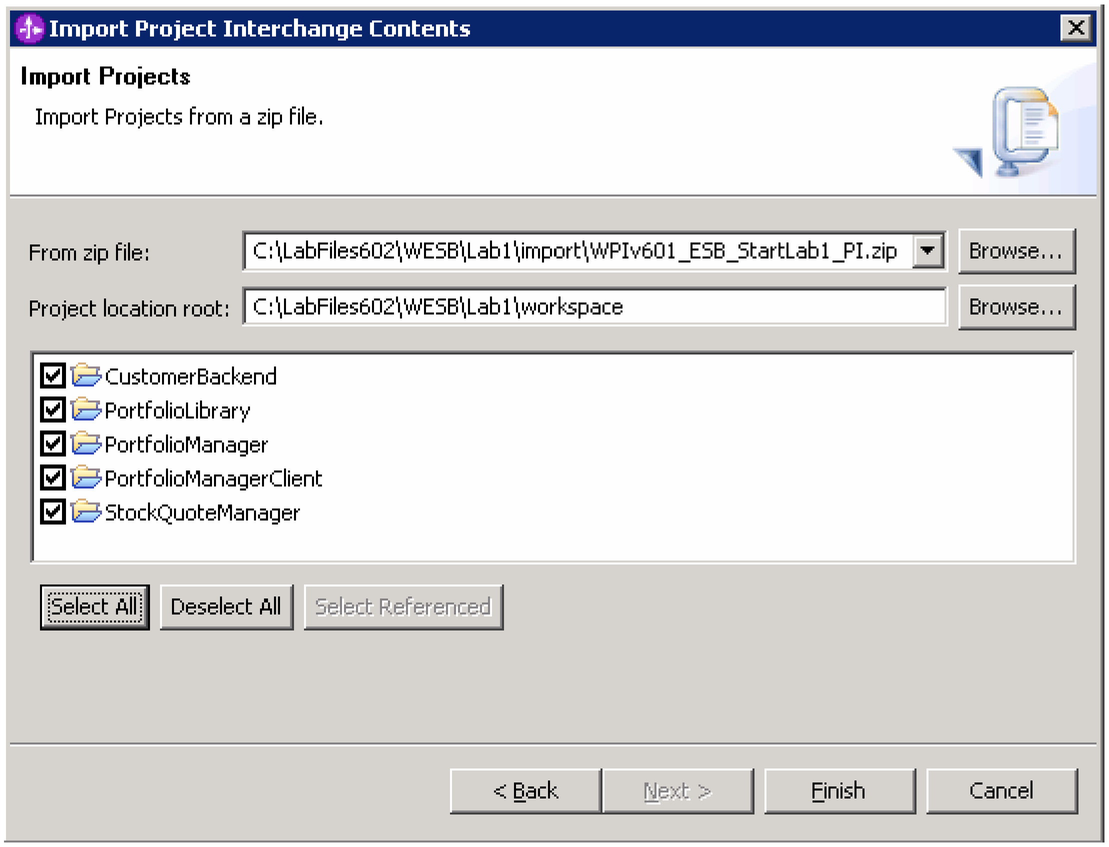
____ 3. Import Project Interchange file, **WPIv602_ESB_StartLab1_PI.zip**, into the development environment.

__ a. Right-click inside **Business Integration View** (top left view in the Business Integration Perspective) and select **Import** from the context menu.



__ b. Select **Project Interchange** from the list. Click **Next**.

__ c. Click the **Browse** button for "**From zip file**" and navigate to **C:/LabFiles602/WESB/Lab1/import/WPIv602_ESB_StartLab1_PI.zip** and hit **Open**.


Import Project Interchange Contents

Import Projects

Import Projects from a zip file.

From zip file: C:\LabFiles602\WESB\Lab1\import\WPIv601_ESB_StartLab1_PI.zip

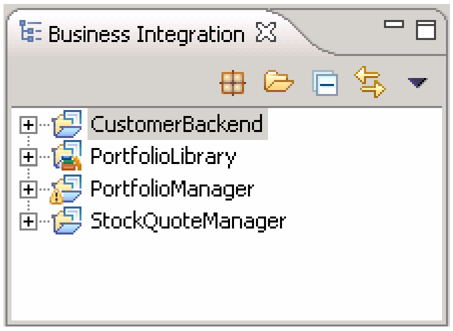
Project location root: C:\LabFiles602\WESB\Lab1\workspace

- [x] CustomerBackend
- [x] PortfolioLibrary
- [x] PortfolioManager
- [x] PortfolioManagerClient
- [x] StockQuoteManager

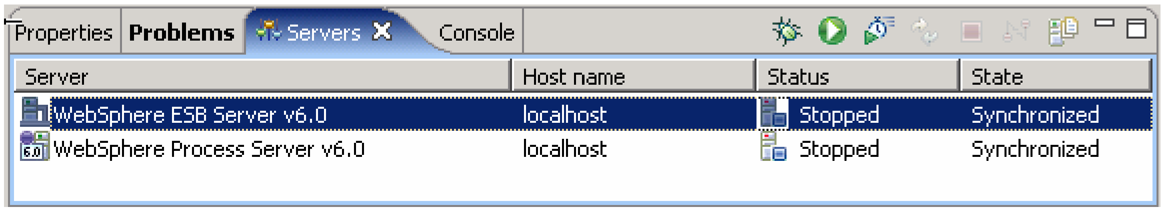
__ d. Click the **Select All** button to select all the checkboxes for the projects listed.

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

__ f. Verify you have **CustomerBackend**, **PortfolioLibrary**, **Portfolio Manager**, and **StockQuoteManager** modules listed in the Business Integration view.



__ g. Verify you have WebSphere ESB Server v6.0 listed in your Servers view.



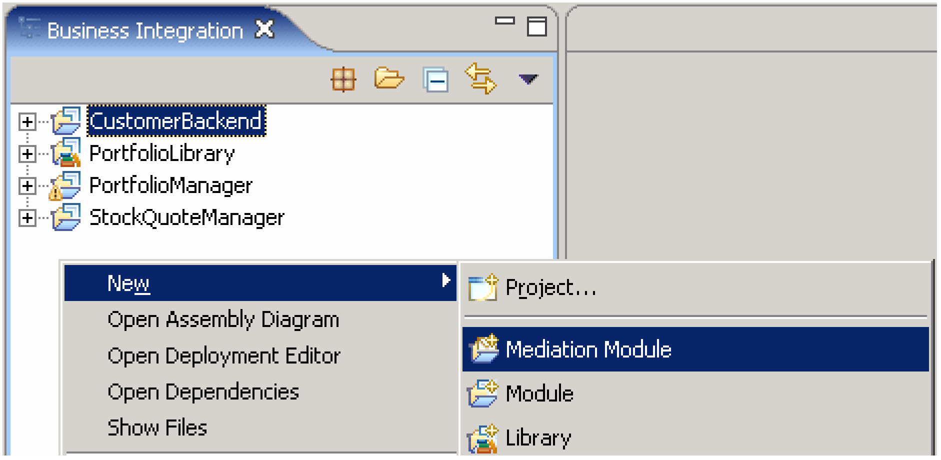
Server	Host name	Status	State
WebSphere ESB Server v6.0	localhost	Stopped	Synchronized
WebSphere Process Server v6.0	localhost	Stopped	Synchronized

Part 2: Create mediation module and mediation flow

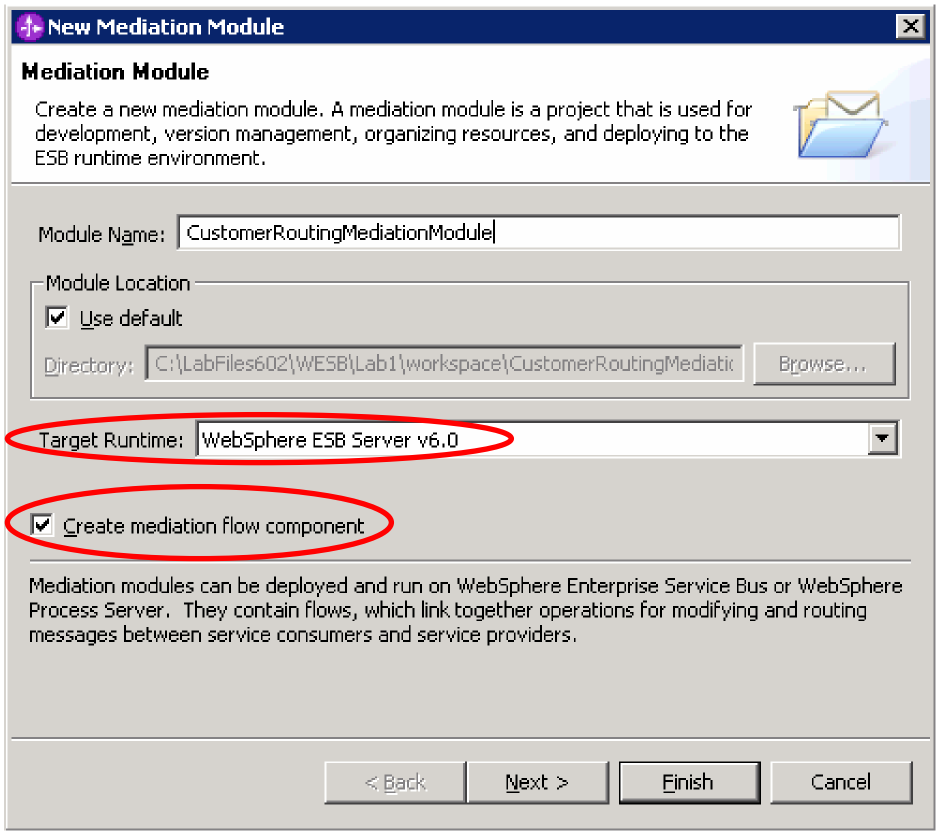
In this section of the lab, you will be creating a new mediation module and a mediation flow. You can only have one mediation module for each deployable project. Every mediation module will have a mediation flow inside of it to connect and control services and mediations.

____ 1. Create a new mediation module.

__ a. Right-click in Business Integration view and select **New → Mediation Module.**



__ b. Name the module: **CustomerRoutingMediationModule.**


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: CustomerRoutingMediationModule

Module Location

[✓] Use default

Directory: C:\LabFiles602\WESB\Lab1\workspace\CustomerRoutingMediati

Target 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.

__ c. Verify target runtime is the WebSphere ESB Server v6.0 and "Create mediation flow component" is selected.

__ d. Click **Next**.

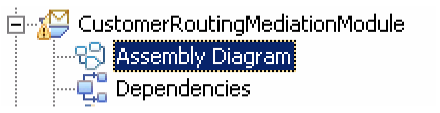
__ e. Select the checkbox for **PortfolioLibrary** to add it as a library project.

__ f. Click **Finish**.

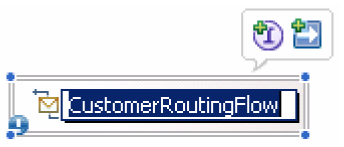
____ 2. Edit Assembly Diagram to prepare Mediation Module.

__ a. Open Assembly Diagram and update Mediation Flow.

1) Expand **CustomerRoutingMediationModule** in the Business Integration view and double click on **Assembly Diagram** (Assembly Diagram) to open it in an Assembly Diagram editor

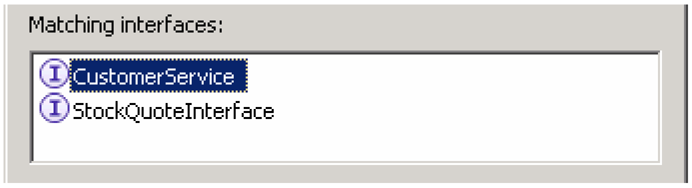


2) Notice that a default mediation flow was created with the name Mediation1. Change the name of the Mediation flow from "Mediation1" to "**CustomerRoutingFlow**".



3) Hover over CustomerRoutingFlow and click the “I” icon ( ) that appears to **add an interface**.

4) Select **Customer Service** from the list of matching interfaces. Click **OK**.


Matching interfaces:

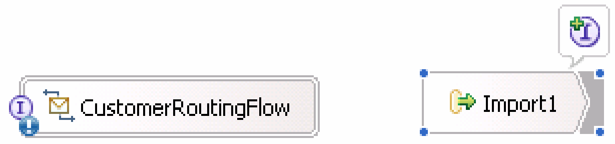
CustomerService
StockQuoteInterface

__ b. Add an Import to **CustomerRoutingFlow** in Assembly Diagram.

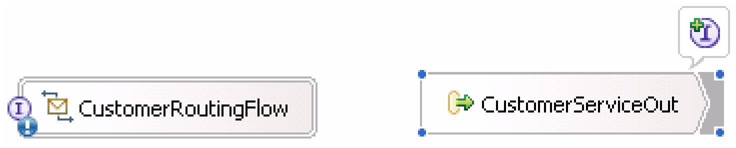
1) Drag and drop an import on the right-hand side of **CustomerRoutingFlow**.

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

b) Click or drag import to the right side of **CustomerRoutingFlow**.



2) Change default import name from Import1 to **CustomerServiceOut**.



3) Hover over import or click on import to make the add interface icon appear.

Click "I" icon ( ) to add an interface.

4) Select **CustomerService** from the list of matching interfaces. Click **OK**.

__ c. Add an Export to CustomerRoutingFlow in Assembly Diagram.

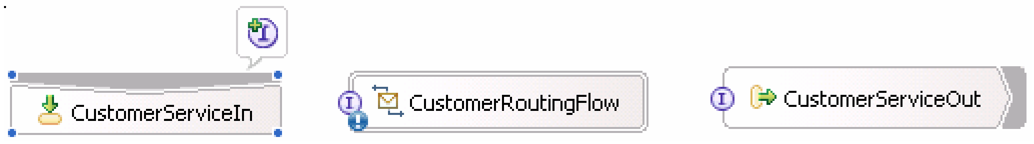
1) Drag and drop an **Export** on the left-hand side of the **CustomerRoutingFlow**.

a) Click on **Export** icon from Assembly Diagram tray ( ).

b) Click or drag icon to left side of **CustomerRoutingFlow**.

2) Change name from Export1 to **CustomerServiceIn**.

.



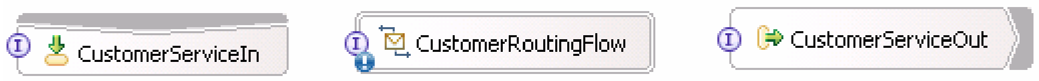
3) Hover over export or click on Export to make the add interface icon appear.

Click "I" icon ( ) to add an interface.

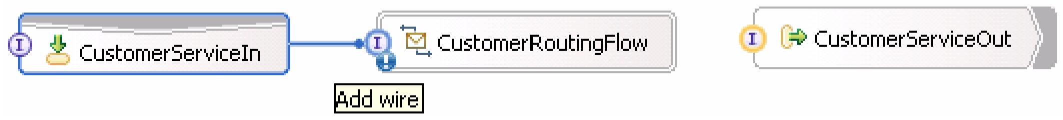
4) Select **CustomerService** from the list of matching interfaces. Click **OK**.

__ d. Wire components together.

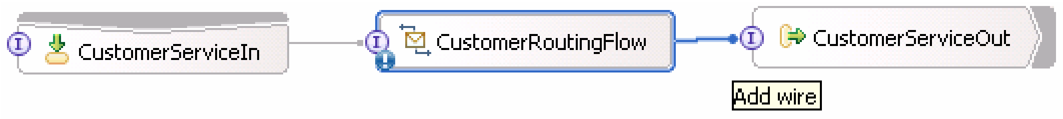
1) Your Assembly Diagram should look like the one below.



2) Click on **CustomerServiceIn** and drag a wire to **CustomerRoutingFlow**.



3) Click on CustomerRoutingFlow and drag a wire to CustomerServiceOut.



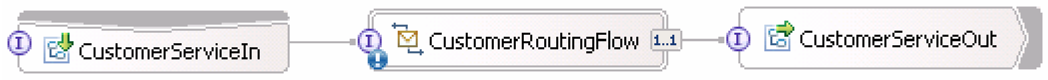
4) Click **OK** to any pop-up windows.

__ e. Generate SCA bindings for Import and Exports.

1) Right-click on CustomerServiceOut and select **Generate Binding → SCA Binding**.

2) Right-click on CustomerServiceIn and select **Generate Binding → SCA Binding**.

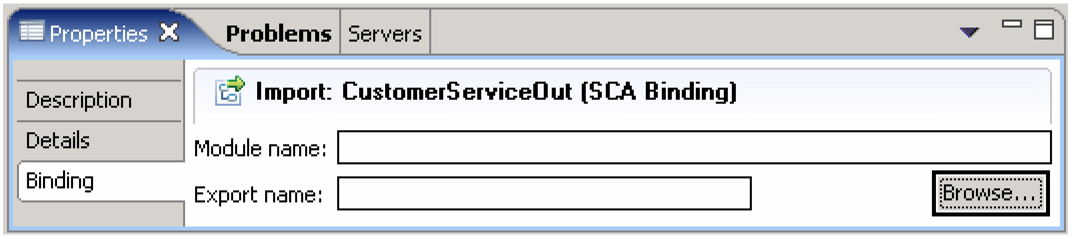
3) This is what your Assembly Diagram should now look like.



__ f. Update binding for CustomerServiceOut.

1) Select the **Import** CustomerServiceOut on the Assembly Diagram.

2) Open Properties view to **Binding tab**.

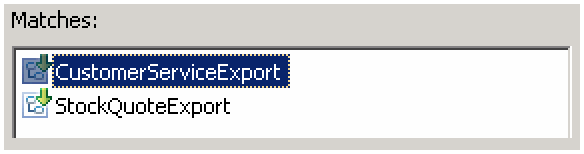


Import: CustomerServiceOut (SCA Binding)

Module name:

Export name:

3) Click browse button and select **CustomerServiceExport**.



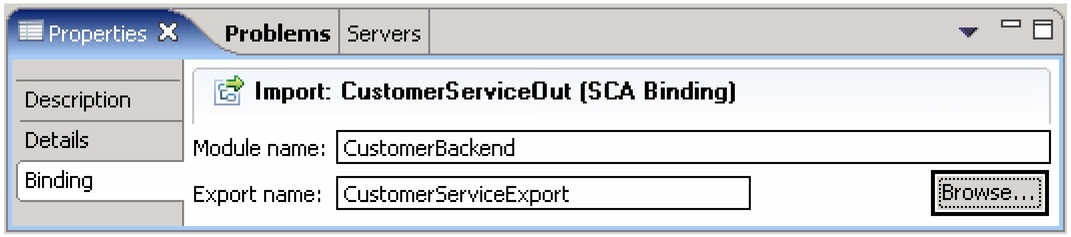
Matches:

CustomerServiceExport

StockQuoteExport

4) Click OK.

5) This is how the binding information for the CustomerServiceOut import looks like



Import: CustomerServiceOut (SCA Binding)

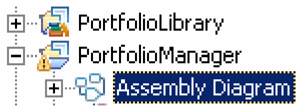
Module name: CustomerBackend

Export name: CustomerServiceExport

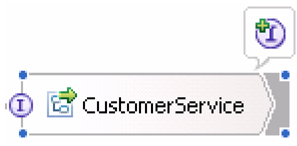
6) Save all work by **File → Save All** or **Crtl + Shift + S**.

__ g. Update import binding "**CustomerService**" for **PortfolioManager** module.

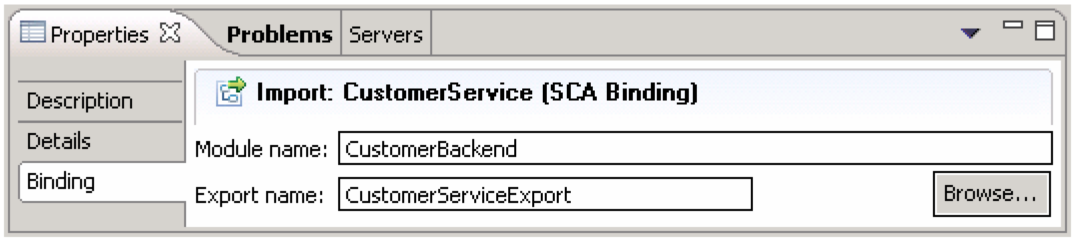
1) Expand **PortfolioManager** in the Business Integration view and double click on **Assembly Diagram** (Assembly Diagram) to open it in an Assembly Diagram editor.



2) Select the import **CustomerService**.



3) Open **Properties View** with the CustomerService import highlighted.
Select the **Binding tab** inside Properties View.

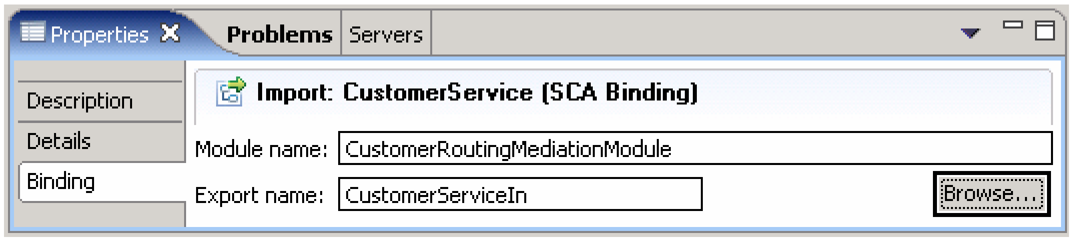


Import: CustomerService (SCA Binding)

Module name: CustomerBackend

Export name: CustomerServiceExport

4) Click **Browse** button and select **CustomerServiceIn** (instead of **CustomerServiceExport** that the binding is set to).



Import: CustomerService (SCA Binding)

Module name: CustomerRoutingMediationModule

Export name: CustomerServiceIn

5) Click **OK**.

6) Save the Assembly Diagram. **File → Save** or **Ctrl + S**.

You have now successfully inserted a mediation flow between the PortfolioManager module and the CustomerBackend module. Redirecting PortfolioManager to **CustomerServiceIn** allows the message to be operated on by the Mediation Flow and then passed to the **CustomerBackend** through the import **CustomerServiceOut**.

Part 3: Generate mediation flow implementation and create a mediation

In this section you will generate the implementation for the Mediation Flow. Afterwards, you will create only one type of mediation; that is, the message logger mediation. It is a simple mediation that sends data (whether the /body, /context, or /header part of a message) to a cloudscape database. In this section, you will also learn how to open a cloudscape database and view the data captured by the message logger.

____ 1. Generate implementation for CustomerRoutingFlow.

__ a. Expand **CustomerRoutingMediationModule** in the Business Integration view and double click on **Assembly Diagram** (Assembly Diagram)

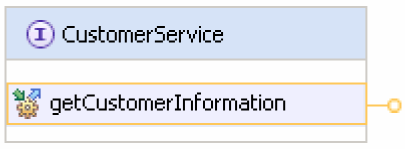
__ b. Right-click on **CustomerRoutingFlow** in the Assembly Diagram for CustomerRoutingMediationModule and select **Generate Implementation** from the context menu Click **OK** over the Generate Implementation dialog box.

NOTE: You can create a mediation flow by right-clicking in the business integration view and select New → Mediation Flow. However, generating the implementation from the assembly diagram created a new Mediation Flow with the name specified in the assembly diagram "CustomerRoutingFlow".

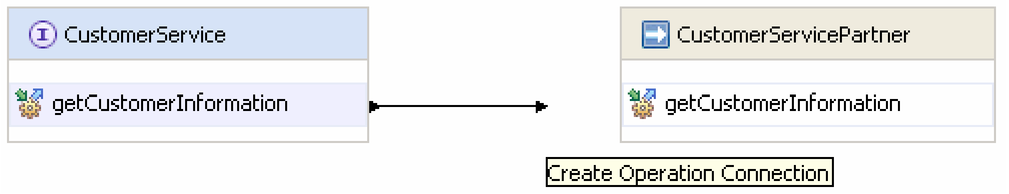
__ c. The **Mediation Flow Editor** will open after generating the mediation CustomerRoutingFlow implementation.

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

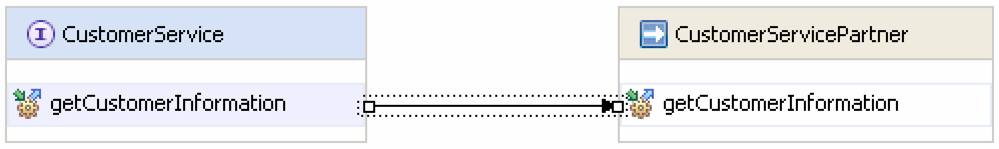
__ a. Click on getCustomerInformation section of the **CustomerService** Interface on the left-hand side of the Operation Connections view.



__ b. Drag to the getCustomerInformation section of the **CustomerServicePartner** on the right-hand side of the Operation Connections view and unclick. There should be a **black line** from CustomerService to CustomerServicePartner.

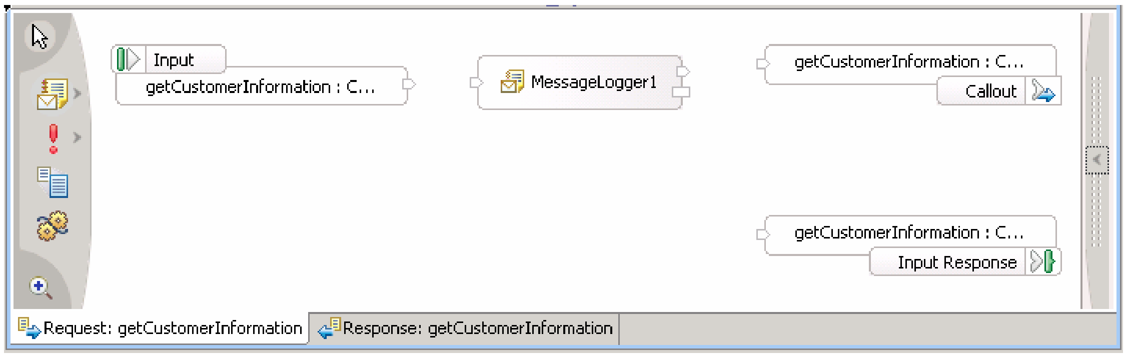


__ c. Click on the **black line** or the **getCustomerInformation** section of the CustomerService Interface (left-hand side) to make the Mediation Flow diagram populate with information (if not already up).



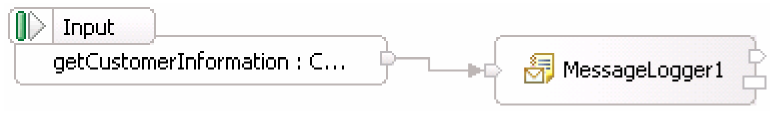
____ 3. Add **Message Logger** mediation to Mediation Flow diagram.

__ a. In **Mediation Flow View** (middle), click on **Message Logger** icon ( ) from the tray on left-hand side and drop/click in the middle of the diagram as shown below:

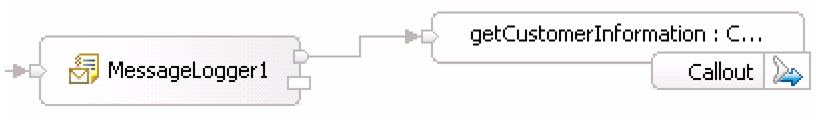

Input
getCustomerInformation : C...
MessageLogger1
getCustomerInformation : C...
Callout
getCustomerInformation : C...
Input Response
Request: getCustomerInformation
Response: getCustomerInformation

____ 4. Wire **input** and **output terminals** for the **Request**.

__ a. Click on the output terminal **"out"** of the getCustomerInformation: CustomerService **Input** node (on top, left-hand side of mediation flow diagram) and drag to the **input terminal** of the message logger.



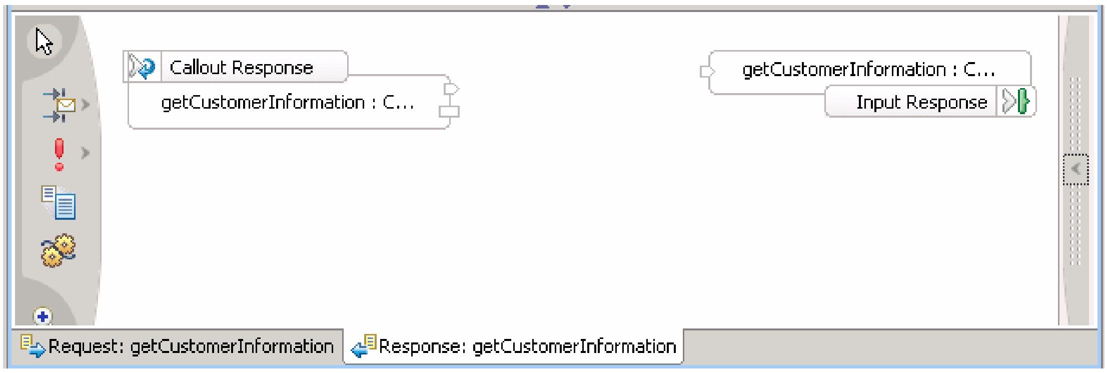
__ b. Click on the **output terminal "out"** of the Message Logger mediation (middle) and drag to the **input terminal** of the getCustomerInformation: CustomerServicePartner **Callout** node (top, right-hand side).



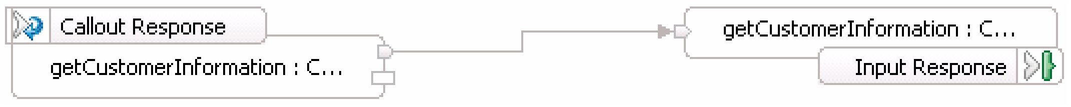
____ 5. Wire **input** and **callout terminals** for the **Response**.

At the bottom of the Mediation Flow view, there are 2 tabs. By default you will see the Request information when enacting the mediation flow view. However, now you need to set up the Response side of the mediation flow. You can set mediations on both sides of the flow.

__ a. Click the **Response: getCustomerInformation** tab.


Callout Response
getCustomerInformation : C...
getCustomerInformation : C...
Input Response
Request: getCustomerInformation
Response: getCustomerInformation

__ b. Click on the output terminal **"out**" of the getCustomerInformation: CustomerServicePartner **Callout Response** node (on the top, left-hand side of mediation flow diagram) and drag to the input terminal of the getCustomerInformation: CustomerService **Input Response** node.



__ c. Save all with **Ctrl + Shift +S** or by navigating to **File → Save All**.

____ 6. Investigate Message Logger.

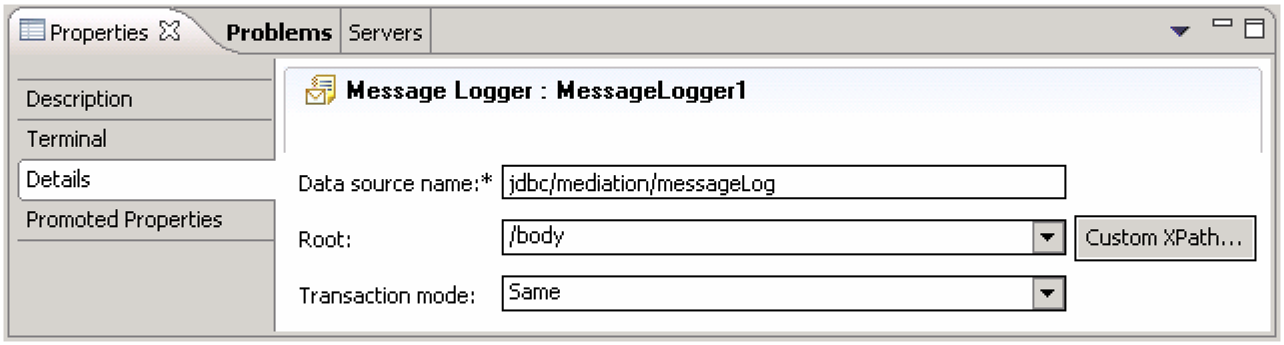
__ a. Click back on **Request tab** at bottom of Mediation Flow Editor.

Request: getCustomerInformation
Response: getCustomerInformation

__ b. Click on **MessageLogger1** mediation.

__ c. Then click on the **Properties View** (lower right quadrant) while MessageLogger1 is selected.

__ d. Click on **Details tab** of Properties view and notice that WebSphere Enterprise Service Bus came with the **data source** to Cloudscape already set up (jdbc/mediation/messageLog).



Message Logger : MessageLogger1

Data source name:* jdbc/mediation/messageLog

Root: /body

Transaction mode: Same

__ e. The "Root" section sets what level of information you will be sent to the database from the message (body, context, or headers). Leave as **/body**.

__ f. Keep the transaction mode at "**Same**". Notice here you have the option of creating a "New" transaction.

Part 4: Test message logger mediation

In this section, you will run the mediation and use CVIEW to see what data was populated into the cloudscape database.

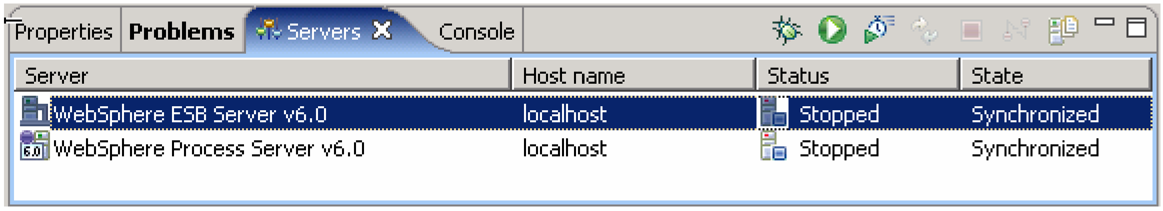
____ 1. Start **WebSphere ESB Server** and **add modules** to server

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

If using a local testing environment:

__ a. Open **Servers View**.

__ b. Select WebSphere ESB Server v6.0 and click **Start button** ( ).



Server	Host name	Status	State
WebSphere ESB Server v6.0	localhost	Stopped	Synchronized
WebSphere Process Server v6.0	localhost	Stopped	Synchronized

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

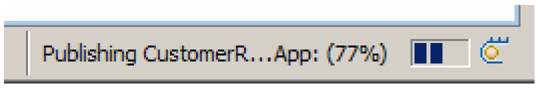
____ 3. Add **projects** to WebSphere Enterprise Service Bus Server (once the ESB Server is started, not before).

__ a. In Servers view, right-click on WebSphere ESB Server v6.0 and select "Add and Remove Projects..."

NOTE: Note that the ESB server you are using is configured with an ESB profile that is part of the installation and not part of the workspace. Therefore, if you have 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. That should avoid any potential errors.

__ b. Click **Add-All>>** button to move all projects to server and click **Finish** button.

__ c. 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.



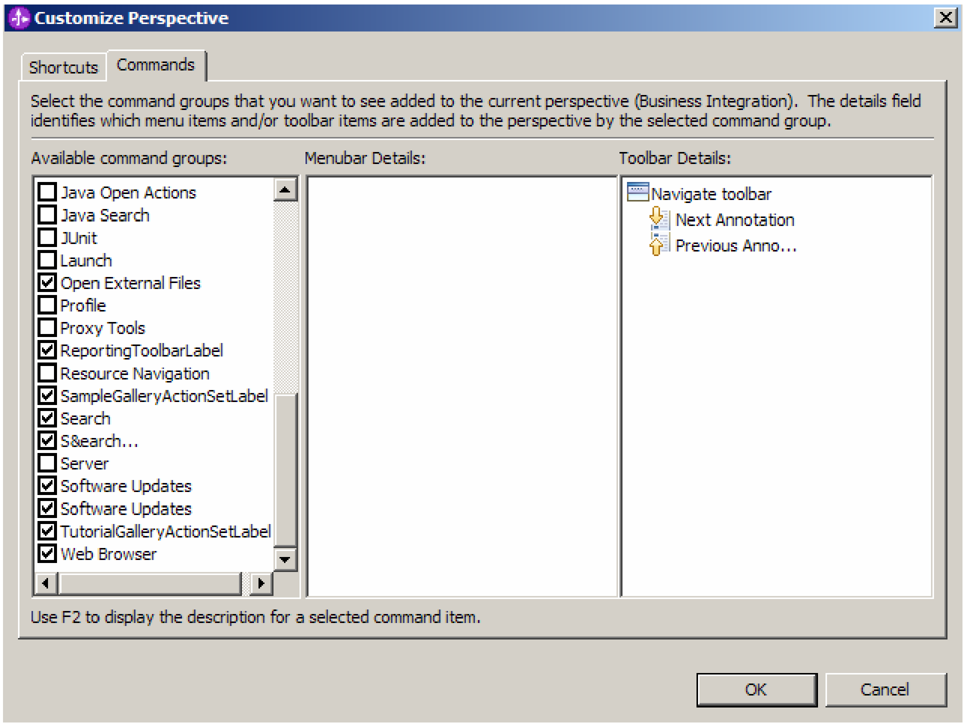
Publishing CustomerR...App: (77%)

____ 4. Enable Web browser icon in WebSphere Integration Developer. You only need to do this once.

__ a. Go to **Window → Customize Perspective**

__ b. Click on the commands tab and scroll to the bottom.

__ c. Click the check box next to Web browser



Select the command groups that you want to see added to the current perspective (Business Integration). The details field identifies which menu items and/or toolbar items are added to the perspective by the selected command group.

Available command groups:

- [ ] Java Open Actions
- [ ] Java Search
- [ ] JUnit
- [ ] Launch
- [✓] Open External Files
- [ ] Profile
- [ ] Proxy Tools
- [✓] ReportingToolbarLabel
- [ ] Resource Navigation
- [✓] SampleGalleryActionSetLabel
- [✓] Search
- [✓] S&earch...
- [ ] Server
- [✓] Software Updates
- [✓] Software Updates
- [✓] TutorialGalleryActionSetLabel
- [✓] Web Browser

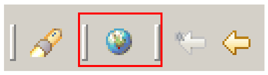
Menubar Details:

Toolbar Details:

Navigate toolbar
Next Annotation
Previous Anno...

Use F2 to display the description for a selected command item.

__ d. Click OK. This should place an icon for Web browser in the WebSphere Integration Developer tools panel.



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

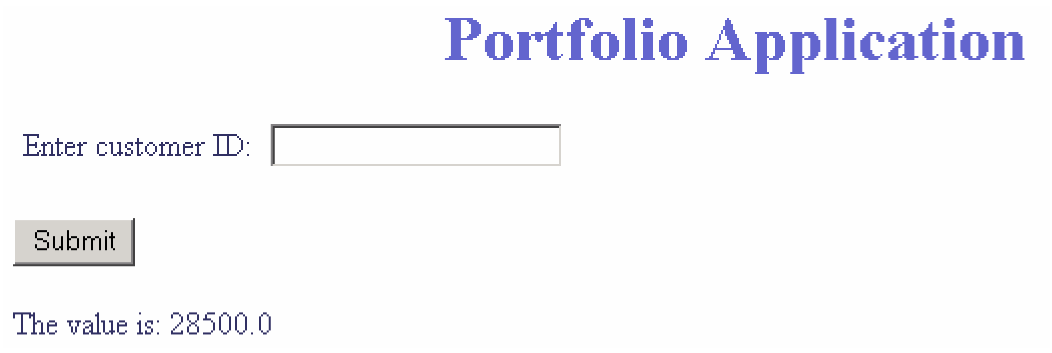
__ f. Enter http:// <HOSTNAME>:<PORT>/PortfolioManagerClient/index.jsp . Where hostname is the name of the system where the WebSphere Enterprise Service Bus server is located. Port is the **WC_defaulthost** port of the WebSphere Enterprise Service Bus profile.

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

Note: You can get the **WC_defaulthost** 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

__ g. Enter **2222222** (2 seven times) in text input box and click **Submit** to get a response displayed to the JSP and to the Console View of WebSphere Integration Developer. You should see "The value is: 28500.0".


Portfolio Application

Enter customer ID:

Submit

The value is: 28500.0

__ h. Close browser.

__ i. Stop WebSphere ESB Server v6.0 by highlighting the WebSphere ESB Server v6.0 in the Servers View and click **Stop** button ( ).
(FYI…You cannot view data in a Cloudscape database when server is running.)

____ 5. Verify data was sent to database by MessageLogger1

If using a remote WebSphere, an option to view Cview data is to start it using X-windows. Another option is to ftp the directory where the database resides to a windows box in binary and then ftp just the service.properties file in ASCII. You can then use the Cview within WebSphere Integration Developer to look at the database.

__ a. Open Cview to view the database **EsbLogMedDB**.

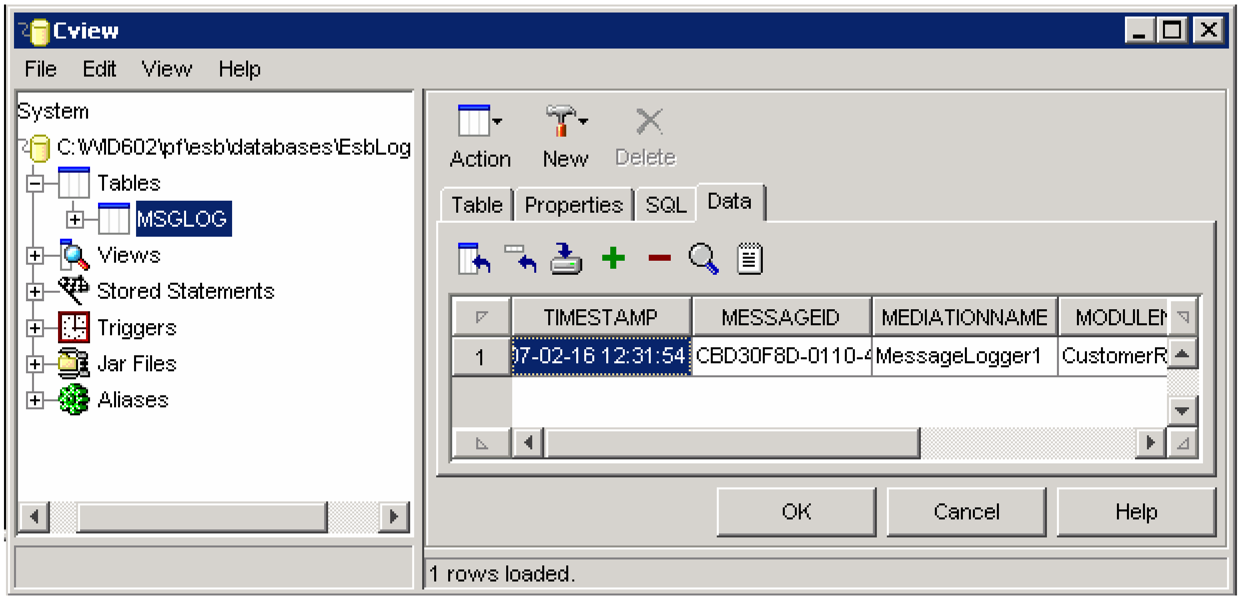
1) Navigate to **<WID_HOME>\runtimes\bi_v6\cloudscape\bin\embedded** and run **cview.bat**.

2) Once the Cview GUI appears, go to **File → Open**.

3) Navigate to **<WID_HOME>\pf\esb\databases\EsbLogMedDB**.

4) Navigate to tables and open **MSGLOG table**.

5) Click on **DATA tab** on the right-hand side of Cview GUI.



	TIMESTAMP	MESSAGEID	MEDIATIONNAME	MODULEN
1	07-02-16 12:31:54	CBD30F8D-0110-4	MessageLogger1	CustomerR

1 rows loaded.

6) There should be one row present. Take scroll bar all the way to the right to view data under the **MESSAGE column**.

7) Click on the **table cell below MESSAGE** to highlight the table cell.

8) Click the **Text Editor** icon (🗒) just above the table, but still under the “Data” tab to view what the message logger saved from the /body of the message. You should see…

```
<?xml version="1.0" encoding="UTF-8"?>
<body xsi:type="service:getCustomerInformationRequestMsg"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:service="http://CustomerBackend/CustomerService">
  <getCustomerInformation>
    <customerID>2222222</customerID>
  </getCustomerInformation>
</body>
```

__ b. Close CView by selecting **File → Exit**

____ 7. You are done with this exercise. Go ahead and clean up the workspace (steps below) for future work.

Part 5: Save work and clean up server

Export project as Project Interchange file

____ 1. In WebSphere Integration Developer, Navigate to **File → Export**.

____ 2. Select Project Interchange.

____ 3. Out of all the projects listed, you only need to add a check next to **6 projects**.
CustomerBackend
CustomerRoutingMediationModule
PortfolioLibrary
PortfolioManager
PortfolioManagerClient
StockQuoteManager.

All other projects are generated upon import of the project interchange.

____ 4. Save in C:/LabFiles602/WESB/Lab1/

____ 5. Name the project interchange WPIv602_ESB_FinishedLab1_PI.zip.

What you did in this exercise

In this lab, you were provided with an initial understanding of how to create a mediation module and mediation flow in WebSphere Integration Developer V6.0.2. You then learned how to run the Message Logger mediation module on the WebSphere Enterprise Service Bus V6.0 server, and check for results in a cloudscape database.

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/Lab1/solution/WPIv602_ESB_Lab1_PI_Solution.zip**

__ c. Select all the 6 project listed

__ d. Click **Finish** button.

____ 2. Start with **Part 4: Test Message Logger Mediation**.

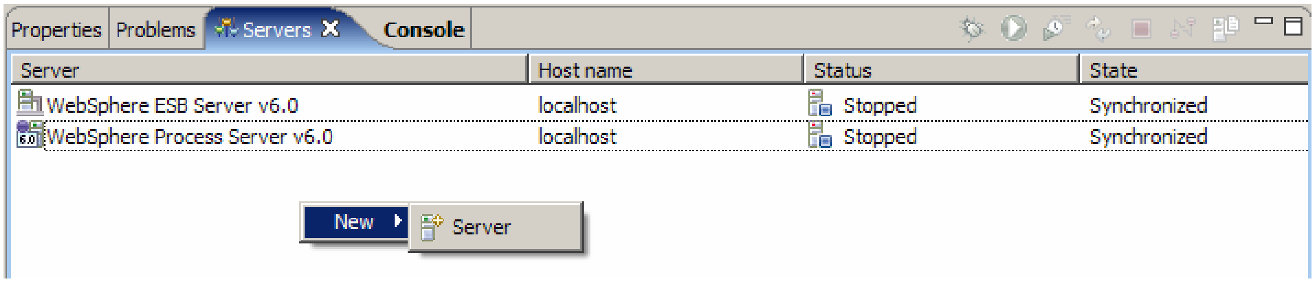
Task: Adding remote server to WebSphere Integration Developer test environment

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.**



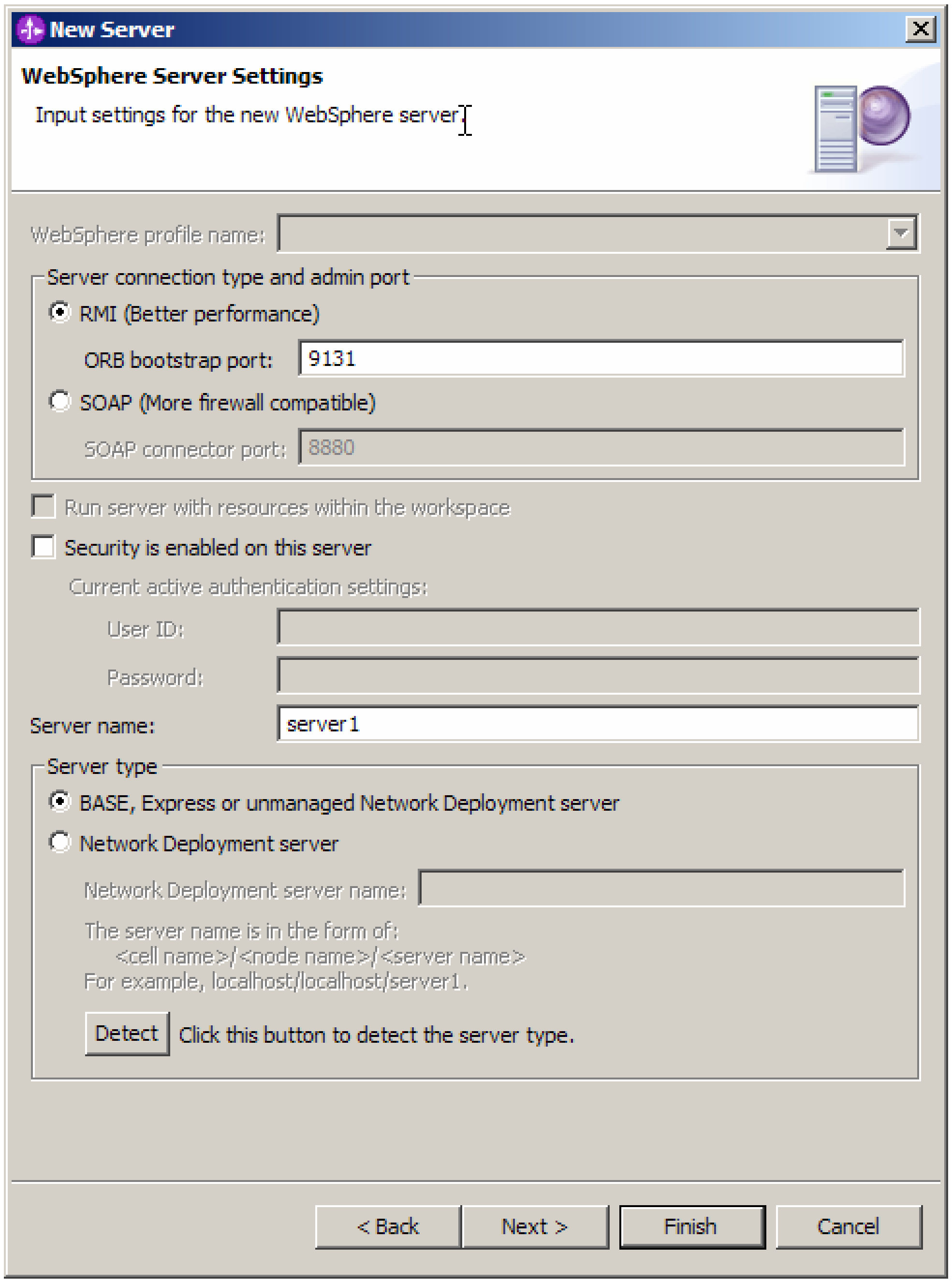
Server	Host name	Status	State
WebSphere ESB Server v6.0	localhost	Stopped	Synchronized
WebSphere Process Server v6.0	localhost	Stopped	Synchronized

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

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

__ e. Click **Next.**

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


New Server

WebSphere Server Settings

Input settings for the new WebSphere server.

WebSphere profile name:

Server connection type and admin port

(●) RMI (Better performance)

ORB bootstrap port: 9131

( ) SOAP (More firewall compatible)

SOAP connector port: 8880

[ ] Run server with resources within the workspace

[ ] Security is enabled on this server

Current active authentication settings:

User ID:

Password:

Server name: server1

Server type

(●) BASE, Express or unmanaged Network Deployment server

( ) Network Deployment server

Network Deployment server name:

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

Detect Click this button to detect the server type.

__ g. Click **Finish**.

____ 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>**

pw : **<PASSWORD>**

If the remote server runs on a System i$^{®}$ server, sign on to the system and start a Qshell session.

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

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

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

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

```
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server cl1sr01 open for e-business; process id is 0000012000000002
```