IBM WebSphere® 6.0 – Lab Exercise

SCA and Web Services

What this exercise is about

This exercise will demonstrate the steps necessary to expose an SCA component as a Web service and invoke that Web service from another SCA module.

Lab Requirements

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

- WebSphere Integration Developer V6.0 installed

- WebSphere Process Server V6.0 test environment installed

What you should be able to do

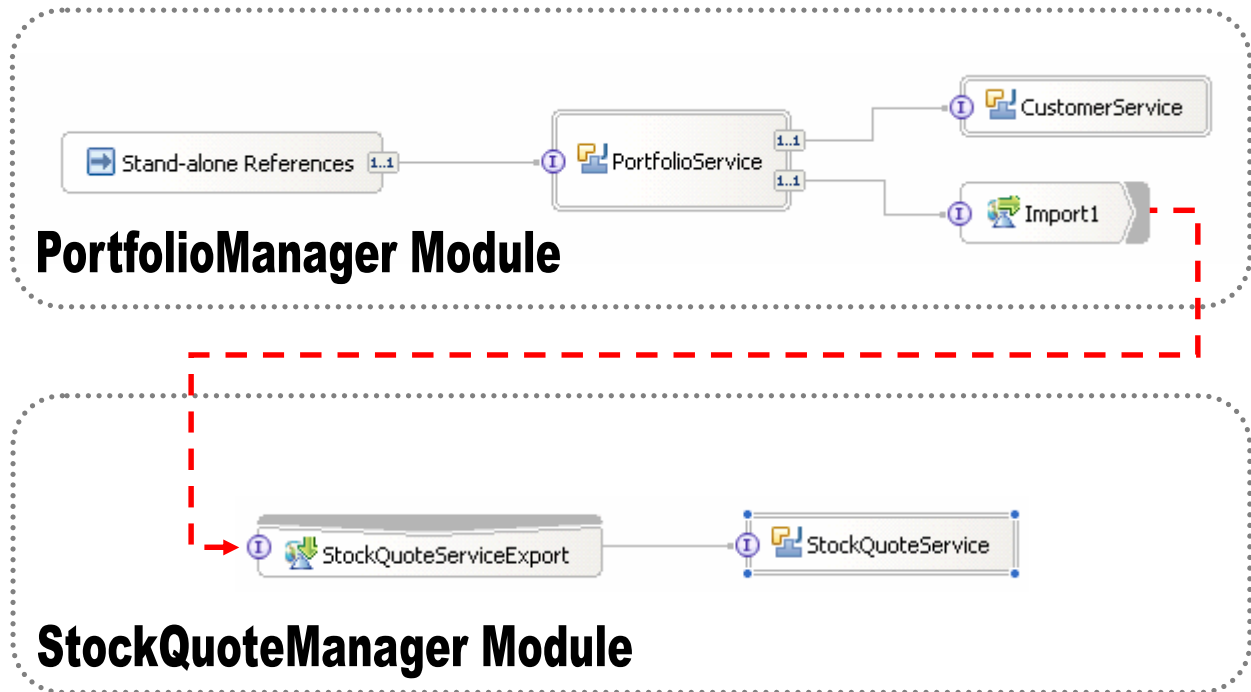At the end of this lab you should be able to:

- Import a project using the Project Interchange feature

- Export an SCA component using a Web Services binding

- Use the Web Service Explorer to test a Web Service

- Invoke a Web Service using a service import

Introduction

The following is an overview of the modules in this exercise:

Exercise Instructions

Some instructions in this lab might be specific for Windows[®] platforms. If you run the lab on a platform other than Windows, you will need to run the appropriate commands, and use appropriate files (for example .sh in place of .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references as follows:

Reference Variable	Windows Location	Linux® Location
<LAB_NAME>	BrokerageWithWS	BrokerageWithWS
<WID_HOME>	C:\Program Files\IBM\WebSphere\ID\6.0	/opt/IBM/WebSphere/ID/6.0
<WPS_HOME>	< WID_HOME >\runtimes\bi_v6	<WID_HOME>/runtimes/bi_v6
<LAB_FILES>	C:\Labfiles60\	/tmp/Labfiles60
<WORKSPACE>	C:\Labfiles60\BrokerageWithWS\workspace	/tmp/Labfiles60/BrokerageWithWS\worksp ace
<TEMP>	C:\temp	/tmp
<SOLUTION>	C:\Labfiles60\BrokerageWithWS\solution	/tmp/Labfiles60/BrokerageWithWS/solution

Windows users: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, you must replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles60\ would be replaced by C:/LabFiles60/

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

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

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

Part 1: Setup Development Environment

_____ 1. Start WebSphere Integration Developer V6 with a new workspace.

    __ a. Select **Start > Programs > IBM WebSphere > Integration Developer V6.0 > WebSphere Integration Developer V6.0** from the start menu.

   __ b. Enter **<LAB_FILES>\<LAB_NAME>\workspace** for your workspace and click **OK** when prompted.
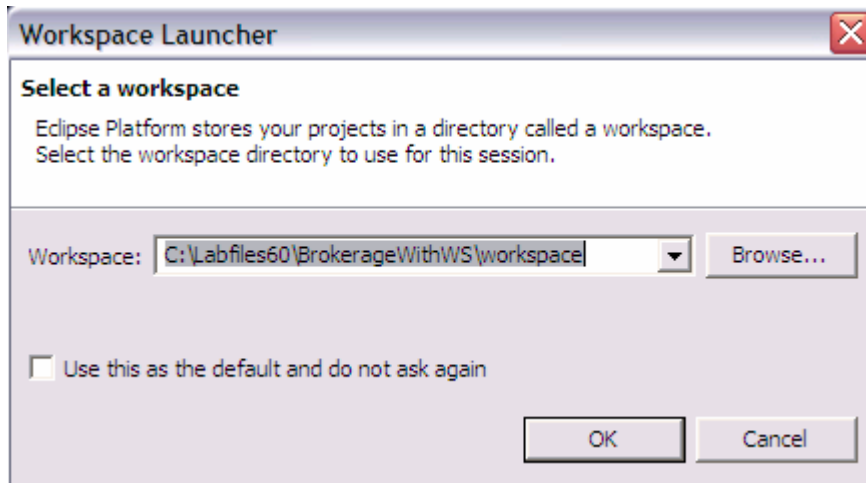
    __ c. Close the **Welcome page** after WebSphere Integration Developer V6.0 opens,.
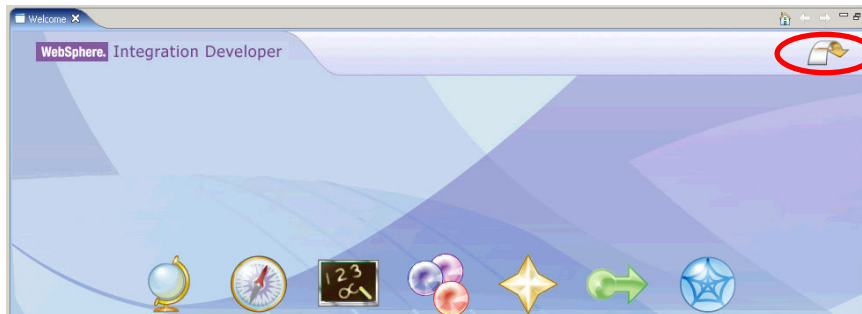
# Part 2: Import PortfolioManager and StockQuoteManager Projects

_____ 1. Import the project interchange file containing the initial code for the PortfolioManager and StockQuote applications.

__ a. Select **File > Import**.

__ b. Select **Project Interchange** from the Import dialog as the import source and click **Next**.

__ c. Use the Browse button to specify the following zipped file:
<LAB_FILES>\<LAB_NAME>\import\BrokerageInitial.zip.

__ d. Enter **<LAB_FILES>\<LAB_NAME>\workspace** for the Project location root.

__ e. Click the **Select All** button to select all projects included in the project exchange .zip file.
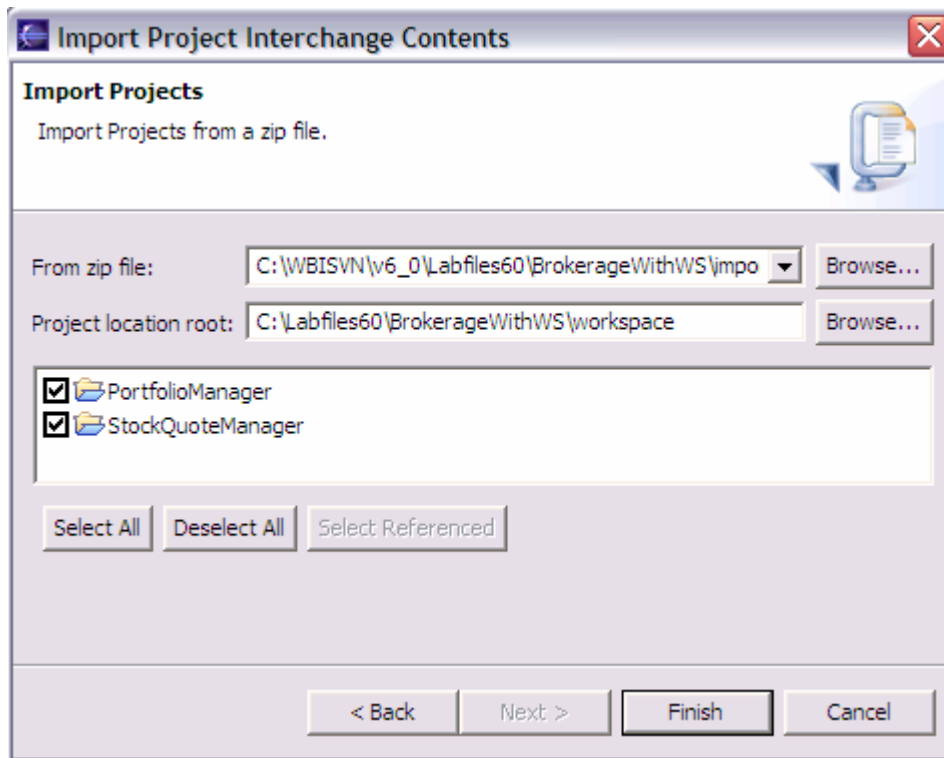
__ f. Click **Finish**.

Part 3: Export the StockQuoteService as a Web Service

_____ 1. Open the assembly diagram for **StockQuoteManager**.

    __ a. Expand **StockQuoteManager** in the Business Integration view and double-click on **StockQuoteManger**.

_____ 2. Export the **StockQuoteService** as a Web Service.

    __ a. In the assembly diagram, right click on the **StockQuoteService** component and select **Export > Web Service Binding**.

    __ b. Click **Yes** in the Binding File Generation dialog box.
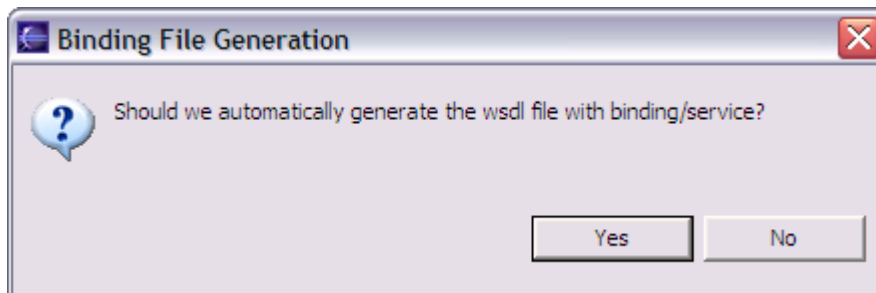
    __ c. Select **soap/http** on the Select Transport dialog and click **OK**.

    __ d. Verify that the assembly diagram looks like the following.

    __ e. **Save** and **close** the assembly diagram.

    __ f. Go to the Window → Show View and click on the Navigator

    __ g. In the Navigator view, expand the StockQuoteManager module

    __ h. Select the StockQuoteServiceExport_StockQuoteInterfaceHttp_Service.wsdl file

    __ i. Right click and select Open with → XML Source Page Editor

    __ j. Delete the localhost in the following line and replace it with the hostname of the remote WebSphere Process Server machine. Also make sure that the port being used corresponds to the remote WebSphere Process Server.

Example:

Change <soap:address location=
"http://**localhost**:9080/StockQuoteManagerWeb/sca/StockQuoteServiceExport"/> to

<soap:address location=
"http://**mvsxxx.rtp.raleigh.ibm.com**:9080/StockQuoteManagerWeb/sca/StockQuoteServiceExport"/
>

Part 4: Test the StockQuoteService using the Web Services Explorer

_____ 3.   Start the server.

    __ a. If using a remote testing environment, follow the directions provided in **Task: Adding Remote Server to WebSphere Integration Developer Test Environment** (at the end of this document) to add a server to the WebSphere Integration Developer test environment and start it. This is especially true for z/OS, AIX, Solaris remote test environment, where the WebSphere Integration Developer will be remote to the test environment.

    If using a local testing environment, right click on the WebSphere Process Server V6.0 from the Server view and select **Start** from the context menu.

    __ b. Wait for the server to start.

_____ 4.   Add the **StockManager** project to the configured projects on the server.

    __ a. Right click on the WebSphere Process Server V6.0 from the Server view and select **Add and remove projects…** from the context menu.

    __ b. Select **StockQuoteManagerApp** from the available projects list and click **Add**.

    __ c. Click **Finish**.

_____ 5.   Enable the Web Services Development capability.

    __ a. Select **Window > Preferences** from the menu.

    __ b. Expand **Workbench** and click on **Capabilities**.

    __ c. Check the box next to **Web Service Developer** in the capabilities list and click **OK**.

_____ 6.   Launch the Web Services Explorer for the **StockQuoteService** Web service.

    __ a. Open the **Physical Resources** view by selecting **Window > Show View > Others** and then expand the **Bussiness Integration** and select **Physical Resources** from the menu. Click **OK**.

    __ b. Expand **StockQuoteManager** and right click on the file **StockQuoteServiceExport_StockQuoteInterfaceHttp_Service.wsdl** and select **Web Services > Test with Web Service Explorer** from the context menu.

__ c. Verify that the Web Service Explorer opens in your work area.
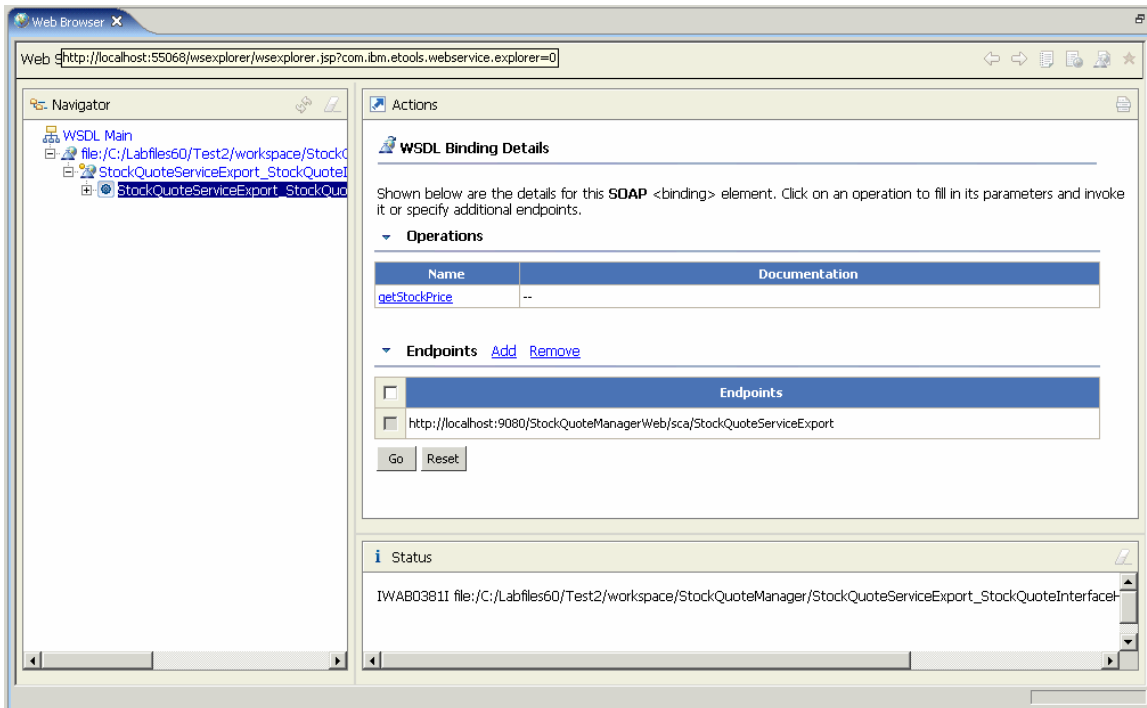
__ d. Click on the **getStockPrice** link to test this operation.


__ e. Enter **IBM** for the symbol and click **Go**.

__ f. Verify that you see the following result in the status window at the bottom of the Web Service Explorer.

__ g. Switch back to **Business Integration** view by selecting **Window > Show View > Bussiness Integration**.

Part 5: Import StockQuoteService into PortfolioManager as a Web Service

_____ 1.   Open the assembly diagram for **PortfolioManager**.

    __ a. Expand **PortfolioManager** in the Business Integration view and double-click on **PortfolioManager**.

_____ 2.   Import the **StockQuoteService** using a web service binding.

    __ a. Locate the **StockQuoteInterface** in the Business Integration view under **PortfolioManager > Interfaces**.

    __ b. Drag and drop the **StockQuoteInterface** on the **PortfolioManager** assembly diagram.

    __ c. Select **Import with Web Service Binding** when the component creation dialog appears and click **OK**.
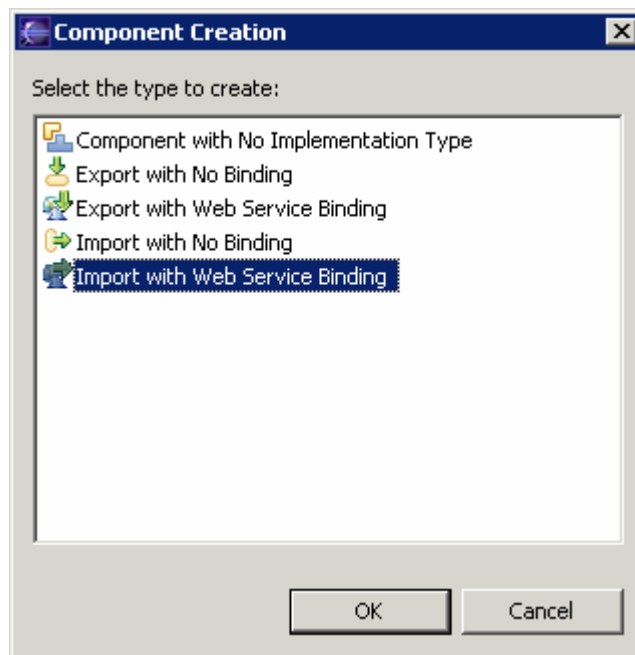
_____ 3.   Create a wire between the **PorfolioService** component and the **Import1** component.

    __ a. Select the wire tool (🔗) and then click on the PorfolioService.

    __ b. Click on the Import1 component.

    __ c. Click **OK** when you see the message appear indicating that a matching reference will be created on the source node.

__ d. Click **No** when prompted if you would like to convert the WSDL interface to a Java interfaces.
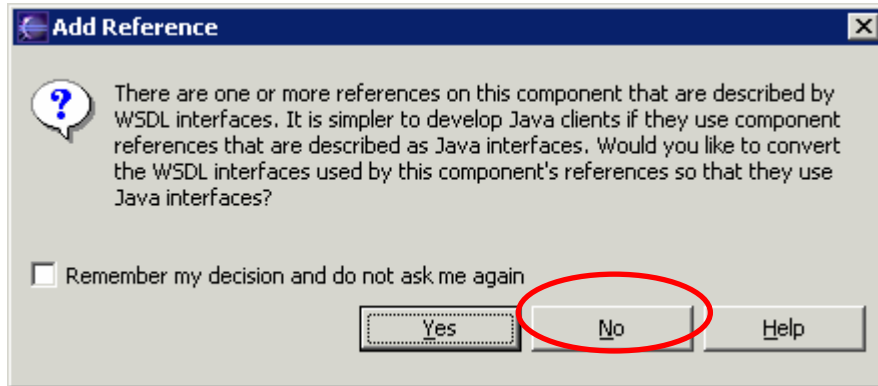
_____ 4.   Verify that your diagram looks like the following:

_____ 5.   Copy the **StockQuoteServiceExport_StockQuoteInterfaceHttp_Service.wsdl** file into the PortfolioManager project.

__ a. Locate the **StockQuoteServiceExport_StockQuoteInterfaceHttp_Service.wsdl** file in the Physical Resources view under StockQuoteManager.

__ b. Right click and select **Copy** from the context menu.

__ c. Remain in the Physical Resources view and right click on the **PortfolioManager** project and select **Paste** from the context menu.

_____ 6.   Update the bindings for Import1.

__ a. Click on Import1 in the assembly editor and go to the binding tab in the Properties view.

__ b. Click the Browse button Next to Service and select **StockQuoteServiceExport_StockQuoteInterfaceHttp_Service.wsdl** from the list.

__ c. Select **StockQuoteServiceExport_StockQuoteInterfaceHttpPort** on the select a port dialog and click **OK**.

_____ 7.   **Save** and **Close** the assembly editor file.

_____ 8.   Import the client JSP file.

__ a. Select **File > Import** from the menu.

__ b. Select 'File System' and click **Next**.

__ c. Click the Browse button from the File system import dialog and choose the following From directory: **<LAB_FILES>\<LAB_NAME>\import**.

__ d. Check the box next to the index.jsp file and click the Browse button to select **PortfolioManagerWeb\WebContent** as the Into folder.
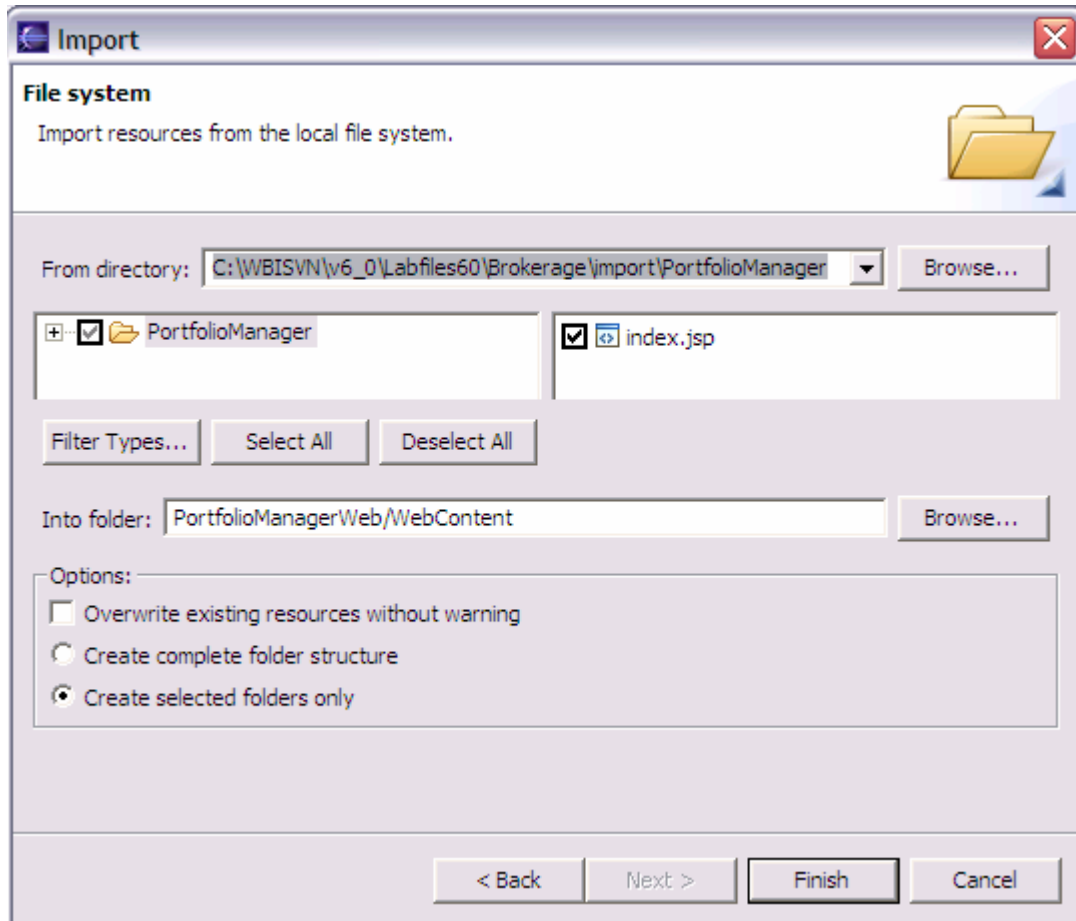
__ e. Click **Finish**.

Part 6: Test the PorfolioManager Application

_____ 1.   Add the PortfolioManager project to the configured projects on the server.

__ a. Right click on the WBI Server V6.0 from the Server view and select **Add and remove projects…** from the context menu.

__ b. Select **PortfolioManagerApp** from the available projects list and click **Add**.

__ c. Click **Finish**.

_____ 2.   Test the PortfolioService service component.

__ a. Open a web browser and enter the following URL.

http://<HOSTNAME>:9080/PortfolioManagerWeb/index.jsp

NOTE: To open the embedded browser within WebSphere Integration Developer V6 select **Window > Customize Perspective** from the menu. Click on the Commands tab in the Customize Perspective dialog and scroll to the bottom of the available command groups list until you find Web Browser. Check the box next to Web Browser and click **OK**. Click the web browser icon (🌐) now in your toolbar to open the embedded web browser.

__ b. Verify that you see the following page displayed in the browser.

Portfolio Application

Enter Customer ID: [   ]

[Submit]

__ c. Enter **123-45-6789** for the Customer ID and click **Submit**.

__ d. Verify that you see the following output.

Portfolio Application

Enter Customer ID: [   ]

[Submit]

The value is: 11500.0

IBM WebSphere 6.0 – Lab Exercise

WPSWIDv6_SCAandWSLab.doc

Part 7: Restore Server Configuration

_____ 1. Remove the StockManager and PortfolioManager project from the configured projects on the server

    __ a. Right click on the WebSphere Business Integration Server V6.0 from the Server view and select **Add and remove projects…** from the context menu.

    __ b. Select PortfolioManager from the configured projects list and click **Remove**.

    __ c. Select StockManager from the configured projects list and click **Remove**.

    __ d. Click **Finish**.

_____ 2. Stop the server.

    __ a. Right click on WebSphere Process Server V6.0 server from the Servers view and select **Stop** from the context menu.

What you did in this exercise

In this exercise you used WebSphere Integration Developer V6 tools to do the following tasks:

- Import a project using the Project Interchange feature

- Export an SCA component using a Web Services binding

- Use the Web Service Explorer to test a Web Service

- Invoke a Web Service using a service import

Solution Instructions

_____ 1.   Start WebSphere Integration Developer V6 with a new workspace.

　__ a. Follow the instructions outlined in Part 1 of this exercise.

_____ 2.   Import the project interchange file **BrokerageWithWS_PI.zip** from
<LAB_FILES>\<LAB_NAME>\solution directory.

　__ a. Select **File > Import** from the menu.

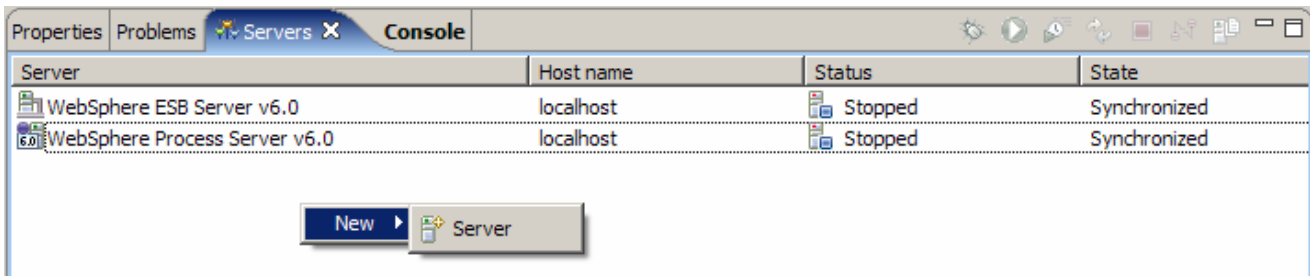　__ b. Select **Project Interchange** in the **Import** dialog and click **Next**.
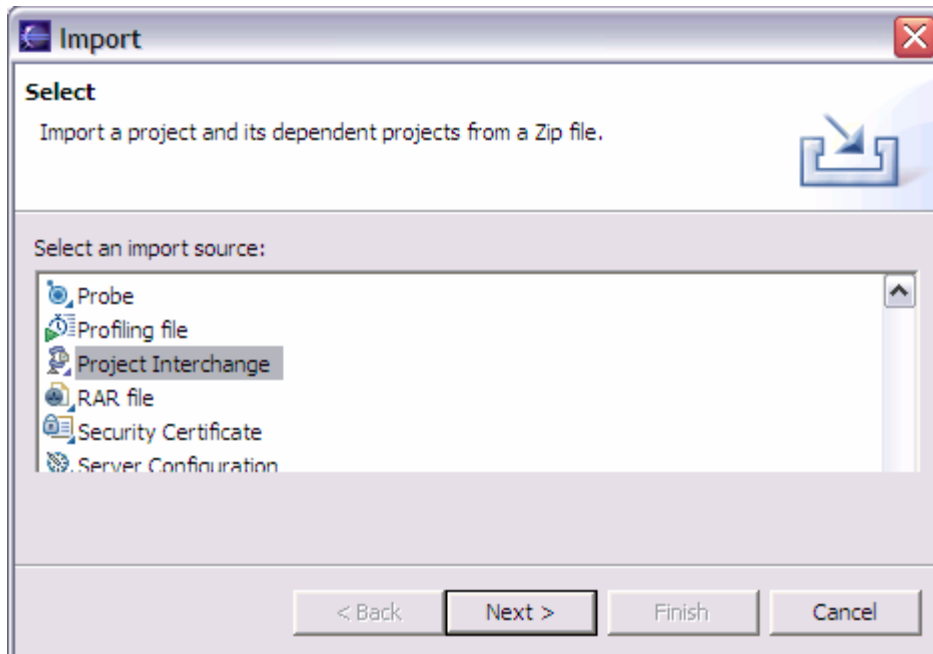
　__ c. Click on the **Browse** button and select the **BrokerageWithWS_PI.zip** in the
<LAB_FILES>\<LAB_NAME>\solution director for the **From zip file**.

　__ d. Enter **<LAB_FILES>\<LAB_NAME>\workspace** for the **Project location root** and click
Next.

　__ e. Click the Select All button and click **Finish**.

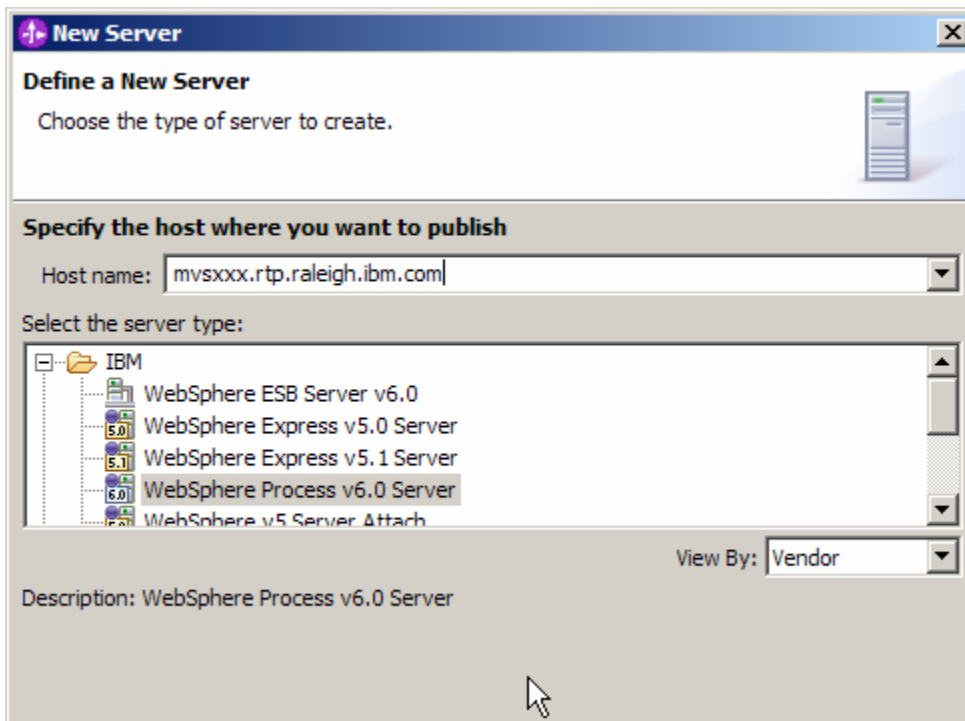_____ 3.   Continue with the lab and do Part 4, 6, and 7 of this lab.

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.

_____ 2.   Right click on the background of the Servers view to access the pop-up menu.

_____ 3.   Select **New > Server.**

Properties	Problems	Servers ✖	Console			
Server			Host name	Status		State
WebSphere ESB Server v6.0			localhost	Stopped		Synchronized
WebSphere Process Server v6.0			localhost	Stopped		Synchronized

New ▶ Server

_____ 4.   Specify hostname to the remote server, **<HOSTNAME>**.

_____ 5.   Ensure that '**WebSphere Process v6.0 Server**' is highlighted in the server type list.

New Server

Define a New Server

Choose the type of server to create.

Specify the host where you want to publish

Host name: mvsxxx.rtp.raleigh.ibm.com

Select the server type:

```
IBM
    WebSphere ESB Server v6.0
    WebSphere Express v5.0 Server
    WebSphere Express v5.1 Server
    WebSphere Process v6.0 Server
    WebSphere v5 Server Attach
```

View By: Vendor

Description: WebSphere Process v6.0 Server

_____ 6.   Click **Next.**

____ 7. On the WebSphere Server Settings page, select the radio button for **RMI** and change the ORB bootstrap port to the correct setting (**<BOOTSTRAP_PORT>**).

____ 8. Click **Finish**.

____ 9. The new server should be seen in the Server view.
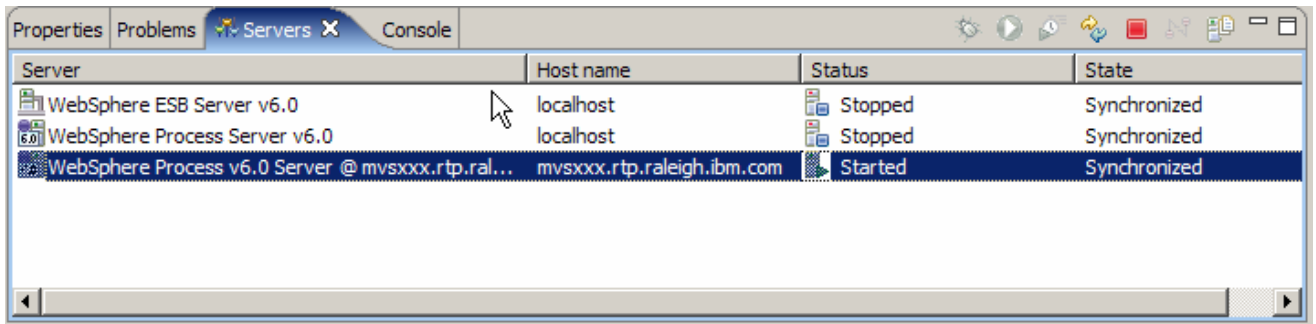
____ 10. Start the remote server if it is not already started. WebSphere Integration Developer does not support starting remote servers from the Server View.

____ 11. From a command prompt, telnet to the remote system if needed:

'**telnet <HOSTNAME> <TELNET_PORT>**'

userid : **<USERID>**

pw : **<PASSWORD>**

____ 12. Navigate to the bin directory for the profile being used:

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

____ 13. Run the command file to start the server: **./startServer.sh <SERVER_NAME>**

____ 14. 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
```

This page is intentionally left blank.

WPSWIDv6_SCAandWSLab.doc