WebSphere Enterprise Service Bus Lab 3
Use the Visual Debugger

What this exercise is about

The objective of this lab is to learn how to prepare an environment for debugging a mediation flow, and then to run the debugger and see what happens to the message as it passes through the mediation flow.

Lab Requirements

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

- WebSphere Integration Developer V6.0.1 with the WebSphere Enterprise Service Bus test server option installed.

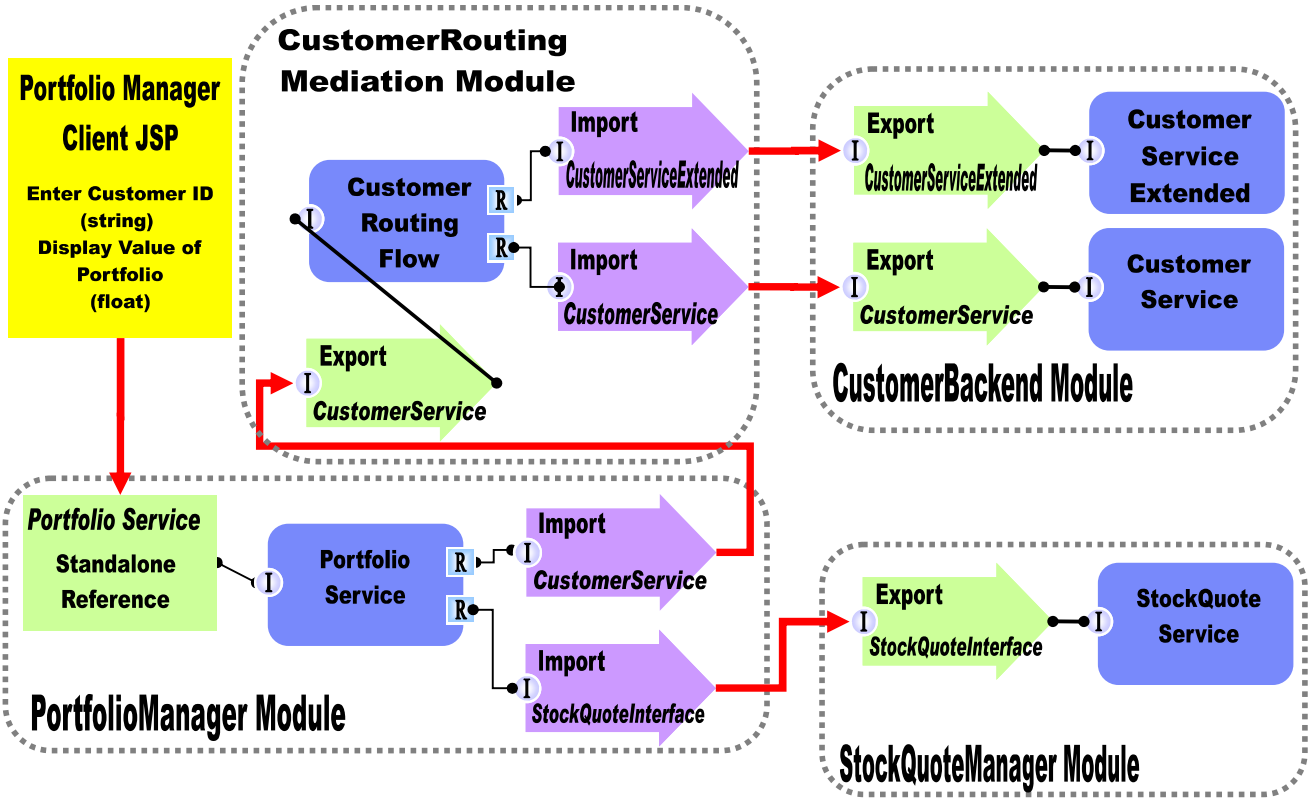
What you should be able to do

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

- Import project interchange files into the WebSphere Integration Developer V6.0 development environment.
- Debug mediation primitives and understand what is happening as a message passes through a mediation flow.
- Set Breakpoints on mediation flow components.

Introduction

In Lab 3 of this lab series, you are going to use the Visual Debugger to step through the above application in order to see what is happening to the message as it passes through the mediation flow. You will first import the application from Lab2 or start with a Project Interchange file that has a working application set up. You will first start server in debug mode. Then, you will open the mediation flow and set breakpoints on all the mediation flow components. Once you have the environment set up, you will bring up a web browser and use a JSP to send a customerID to the new backend. After submitting the customerID in the JSP, the debug perspective will appear and allow you to step through the application stopping at every breakpoint. This is useful since you can look at the Service Message Object (SMO) in Variables view and see what is changing. Purple check-marks on the wire will show you which components were successfully passed. The message will traverse through the Message Logger, Custom Mediation, Database Lookup, Message Filter, and XSLTransformation in order to show you what each component does to or reads from the SMO information, for the response and request side of the flow.



Exercise Instructions

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

Reference Variable	Windows Location	AIX®/UNIX® Location
<WID_HOME>	C:\Program Files\WebSphere\IBM\ID\6.0	
<LAB_FILES>	C:\Labfiles601	/tmp/Labfiles601
<TEMP>	C:\temp	/tmp

Windows users note: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles601\ would be replaced by C:/LabFiles601/

Part 1: Prepare Environment for Lab 3

In this section of the lab, you will be importing all projects inside the Lab3 project interchange file into your workspace.

____ 1. Start WebSphere Integration Developer V6.0.1 with a workspace location of **C:\LabFiles601\WESB\Lab3\workspace**.

____ 2. On the Welcome window, click the curved arrow at top right to **go to workbench**.

____ 3. Import Project Interchange file, **WPIv601_ESB_StartLab3_PI.zip**, into the development environment if starting fresh, or **WPIv601_ESB_FinishedLab2_PI.zip** if continuing from Lab2.

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

__ b. Select **Import** from list.

__ c. Select **Project Interchange** from list. Click **Next**.

__ d. Select the top **Browse** button for **From zip file**:

__ e. Navigate to **C:/LabFiles601/WESB/Lab3/ WPIv601_ESB_StartLab3_PI.zip**.

__ f. Click the **Select All** button to check all checkboxes for projects listed.

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

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

____ 4. Start the ESB Server in debug mode.

__ a. Verify you have WebSphere ESB Server V6.0 listed in your Servers view.

__ b. Right click on the ESB Server and select Debug. This will start the server in debugging mode.

____ 5. While the server is starting in debug mode, add breakpoints to the mediation components. You are setting breakpoints on the mediation components to allow the debugger to stop at each component.

Open the CustomerRoutingMediation **mediation flow**.

In the Business Integration view, navigate to **CustomerRoutingMediationModule -> Mediation Logic -> Flows ->** and double-click **CustomerRoutingFlow**.

Click on the **wire** that is connecting CustomerService with CustomerServicePartner and CustomerServiceExtendedPartner. The Mediation Flow will display for the Request side.

Right-click on the input (CustomerService_getCustomerService_input) and select **Add Breakpoint**. Notice the little blue icon that appears on the component once a breakpoint has been added.

Add a breakpoint to all the **mediation primitive components** in the mediation flow (MessageLogger1, CustomMediation1, DatabaseLookup1, MessageFilter1, XSLTransformation1).

Add a breakpoint to the **2 callouts** for CustomerServicePartner and CustomerServiceExtendedPartner.

Click on the **Response tab** at the bottom of the mediation flow.

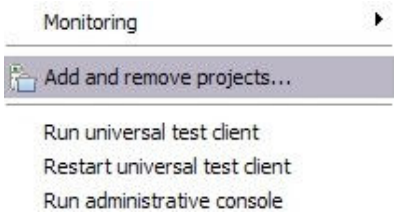
Add a breakpoint to the **2 inputs** for CustomerServicePartner and CustomerServiceExtendedPartner.

Add a breakpoint to the **XLSTransformation1**.

Add a breakpoint to the **input response** for CustomerService.

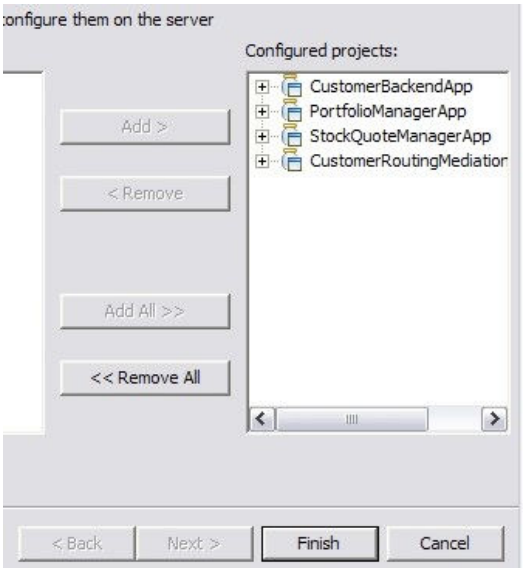
____ 6. Add **projects** to WESB Server (once the ESB Server is started in debug mode, not before).

__ a. In Servers view, right-click on WebSphere ESB Server V6.0 and select **Add and Remove Projects..**.



NOTE: Notice you are using an ESB profile and not a different server. Therefore, if you have projects with the same name deployed to the server, there may be some naming conflicts. Open the administrative console and stop/uninstall those same-named projects before adding these projects to avoid errors.

__ b. Click **Add-All** button to move all projects to server and click **Finish** button. Wait for the deployment to finish



Part 2: Debug Mediation Flow

In this section you will step through all the components from the mediation flow in the debug perspective.

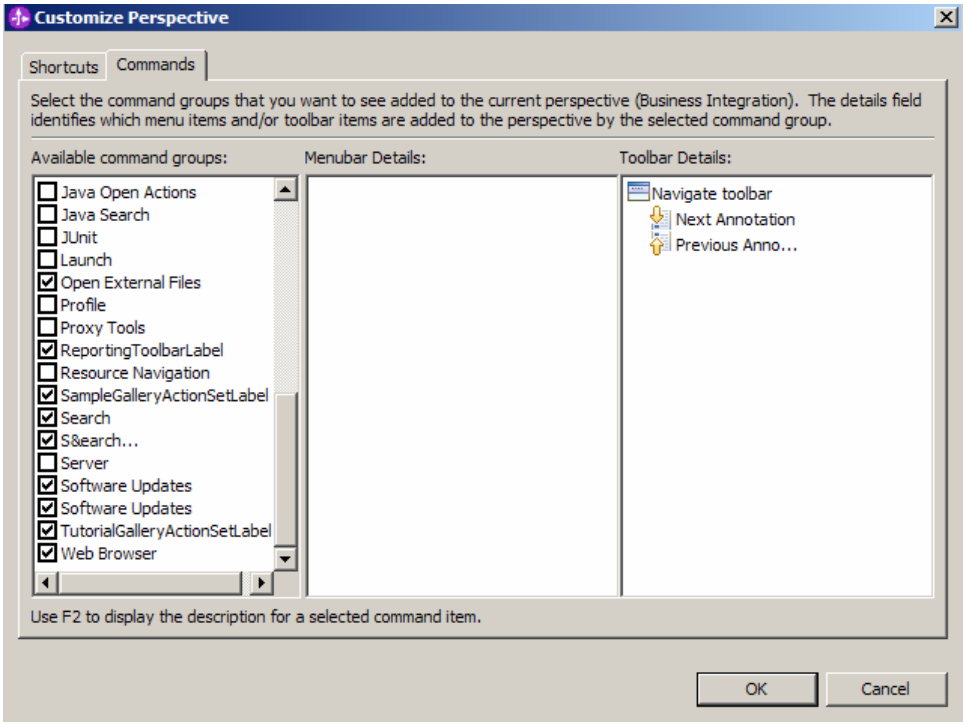
____ 1. Start debugging session by submitting a customerID through the JSP.

Enable web browser icon in WebSphere Integration Developer. You only need to do this once.

1) Go to **Window → Customize Perspective**

2) Click on the commands tab and scroll to the bottom

3) Click the check box next to Web browser



4) Click OK. This should put an icon for Web browser in the WebSphere Integration Developer tools panel.

5) Click on the web browser icon to launch a browser in WebSphere Integration Developer .

6) Enter http://hostname:port/PortfolioManagerClient/index.jsp . Where hostname is the name of the system where the WESB server is located. Port is the **WC_defaulthost** port of the WESB profile.

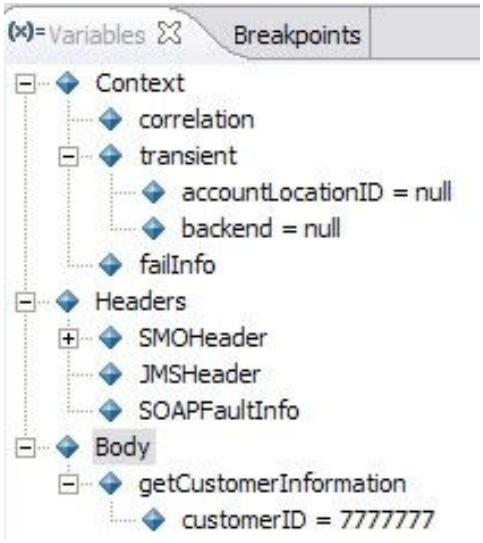
Ex: http://localhost:9080/PortfolioManagerClient/index.jsp

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\WID601\pf\esb\config\cells\esbCell\nodes\

7) Enter **7777777** or 7 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.

__ b. Once you have submitted the customerID, the debug perspective will appear and stop at the first breakpoint on the input.

____ 2. Now in debug perspective, look at the Variables view (top-right corner) and expand the structure of a Service Message Object (SMO).



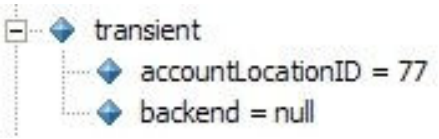
__ a. The first section of the SMO holds context information stored in the correlation and transient contexts, and the failInfo. In this lab series, you created a transient context business object that holds an accountLocationID and backend values (both string variables).
Next is the header information. The SMO has its own header information, along with header information depending upon where the message came from. For example, JMS headers from messaging engines and SOAP headers from a web service.
Then there is the body that holds the actual message information. In this case, the message holds a customerID of 7777777.

____ 3. Click the **Resume** button in top-left hand corner of the debug menu ().

____ 4. Notice there is a purple check-mark on the wire where the debugger has traversed. In this case, you can see that the debugger has successfully moved the message to MessageLogger1.
Click the **Resume** button.

____ 5. Now the debugger is in the CustomMediation1. The custom mediation uses a java snippet that will extract a two digit prefix from the customer ID and place 2 digit it in the accountLocationID field of the transient context. In this case, the value will be 77. Click the **Resume** button.

____ 6. Now that the debugger has traversed through the custom mediation and has reached the DatabaseLookup1, see the transient context in the SMO (Variables View). The accountLocationID should now equal 77 instead of null. Click the **Resume** button.



____ 7. In the Database Lookup primitive, it uses the two digit prefix (77) from the transient context as key to lookup a backend identifier to also 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. So for this example, the two digit prefix will send the message to the CustomerServiceExtended backend. Click the **Resume** button.

____ 8. Check the Variables view to see the transient context in the SMO now has backend = NEW1. Click the **Resume** button.

____ 9. To determine the message routing based on backend identifier, a Message Filter primitive had been added. The old backend will go directly to the callout for CustomerService and the new backend will go to the XSL Transformation in message filter. In this instance, the message will go to the XSL Transformation. Click the **Resume** button.

____ 10. Now in the XSL Transformation, check the Variables view to see the SMO body. The XSL Transformation is going the change the body from the Customer business object to the CustomerExtended business object. Click the **Resume** button.

____ 11. Notice body has changed to CustomerExtended and has gone to the response side. Click on the **Response:getCustomerInformation**. Click the **Resume** button.

____ 12. Check the Variables view to see the SMO body has information for a customer on which shares they own and how many shares they have of each company. Click the **Resume** button.

____ 13. Now in the variables view, see how the SMO body has changed. The application only knows how to communicate with the Customer business object and does not know how to use the CustomerExtended business object. Therefore, this XSL Transformation is taking the body in the form of CustomerExtended business object and transforming it back in to a Customer business object so the application understands. Click the **Resume** button to see this transformation in the body.

____ 14. You would need to hit resume a couple more times till the process completes.

____ 15. You have now traversed the application to see what happens to the SMO while being operated on mediation primitives.

Part 3: Save Work and Clean Up Server

____ 1. Export project as Project Interchange file

__ a. Navigate to **File -> Export**.

__ b. Select **Project Interchange**.

__ c. 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 into a workspace.

__ d. Save in C:/LabFiles601/WESB/Lab3/

__ e. Name the project interchange **WPIv601_ESB_FinishedLab3_PI.zip**.

____ 2. Go ahead and clean up the **ESB Server**.

1. Start WebSphere ESB Server V6.0 from the Servers view of the Business Integration perspective.
2. Right-click on WebSphere ESB Server V6.0 (once started) and select Add and Remove projects...
3. Select Remove-All and click Finish.
4. After remove is done, stop the WebSphere ESB Server V6.0.

What you did in this exercise

In this lab you used the Visual Debugger to step through the above application in order to see what is happening to the message as it passes through the mediation flow. You first imported the application from Lab2 or start with a Project Interchange file that has a working application set up. You first started the server in debug mode. Then, you will open the mediation flow and set breakpoints on all the mediation flow components. Once you had the environment set up, you brought up a web browser and used a JSP to send a customerID to the new backend. After submitting the customerID in the JSP, the debug perspective appeared and allowed you to step through the application stopping at every breakpoint. This was useful since you were able to look at the Service Message Object (SMO) in Variables view and see what was changing. Purple check-marks on the wire showed you which components were successfully passed. The message traversed through the Message Logger, Custom Mediation, Database Lookup, Message Filter, and XSLTransformation in order to show you what each component does to or reads from the SMO information, for the response and request side of the flow.

Solution Instructions

____ 1. No **Solution** since Project Interchange file does not save breakpoint information when exported.

This page is left intentionally blank.