

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

## Log and confidential trace lab

What this exercise is about .....	1
Lab requirements .....	1
What you should be able to do .....	1
Introduction .....	2
Exercise instructions .....	2
Part 1: Initialize the workspace and prepare for the lab.....	5
Part 2: Configure non-pass through inbound scenario .....	6
2.1. Add Java component .....	18
Part 3: Deploy the application and enable tracing .....	20
Part 4: Test the application and review confidential messages.....	23
Solution instructions .....	25
What you did in this exercise .....	27
Task: Adding remote server to WebSphere Integration Developer test environment .....	28

### What this exercise is about

The objective of this lab is to provide you with an understanding of some of new features in WebSphere Adapter for Flat Files V7.0 – usage of Adapter ID for logging and tracing details, hide confidential trace, and poll sub directories in event directory.

### Lab requirements

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

- WebSphere Integration Developer V7.0 installed and updated with latest fixes
- WebSphere Process Server V7.0 Test Environment installed and updated latest fixes
- Extract Labfiles70.zip to your C:\ (your root) drive

### What you should be able to do

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

- Import Flat File adapter RAR file into WebSphere Integration Developer
- Use External Service wizard to configure Activation Spec Properties, Resource Adapter Properties to generate Business Objects

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- Deploy the adapter application onto WebSphere Process Server and enable tracing in WebSphere Process Server administrative console
  - Test the deployed application using WebSphere Process Server test environment
  - Load the trace file and review trace messages from the new Server Logs view
  - Review the trace/log file contents for the 'confidential' messages replaced with 'XXX'
  - Restore the server configuration
- 

## Introduction

The adapter provides the ability to protect sensitive or confidential data in log and trace files from being seen by those without authorization.

Log and trace files for the adapter can contain data, which might contain sensitive or confidential information. Sometimes it is necessary for these files to be seen by individuals without authorization to view to sensitive customer data, for example, a support specialist that must use the log and trace files to troubleshoot a problem.

To protect the data in situations like this, the adapter provides the HideConfidentialTrace property. The HideConfidentialTrace property specifies whether you want to prevent confidential user data from displaying in the adapter log and trace files. When this property is enabled, the adapter replaces the confidential data with XXX's.

The types of information that are considered potentially sensitive data and hidden are:

- The contents of a business object
- The contents of an event record
- User ID
- Business object data in an intermediate form, such as a comma-delimited version of a file

The types of information that are not considered user data and not hidden are:

- Business object schemas
  - Transaction IDs
  - Event IDs
  - Call sequences
- 

## Exercise instructions

Some instructions in this lab are 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

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

appropriate files (.sh or .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference variable	Windows location	AIX® or UNIX® location
<WID_HOME>	C:\Program Files\IBM\WID70	
<WPS_HOME>	C:\<WID_HOME>\runtimes\bi_v70	
<FFADAPTER_HOME>	<WID_HOME>\ResourceAdapters\FlatFile_7.0.0.0\deploy	
<LAB_FILES>	C:\Labfiles70	/tmp/Labfiles70
<WORKSPACE>	<LAB_FILES>\LogAndTrace\workspace	
<EVENT_DIR>	<LAB_FILES>\LogAndTrace\eventdir	
<ARCHIVE_DIR>	<LAB_FILES>\LogAndTrace\archivedir	
<FFFILES>	<LAB_FILES>\FFFfiles	
<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, replace C:\Labfiles70\ with C:/Labfiles70/

---

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

**Instructions if using a remote server for testing**

Note that the previous table is relative to where you are running WebSphere Integration Developer. The table below is related to where you are running the 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	sssr011	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<HOSTNAME>	Localhost	mvsxxx.rtp.raleigh.ibm.com	
<SOAP_PORT>	8880	8880	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	ssadmin	
<PASSWORD>	N/A	fr1day	

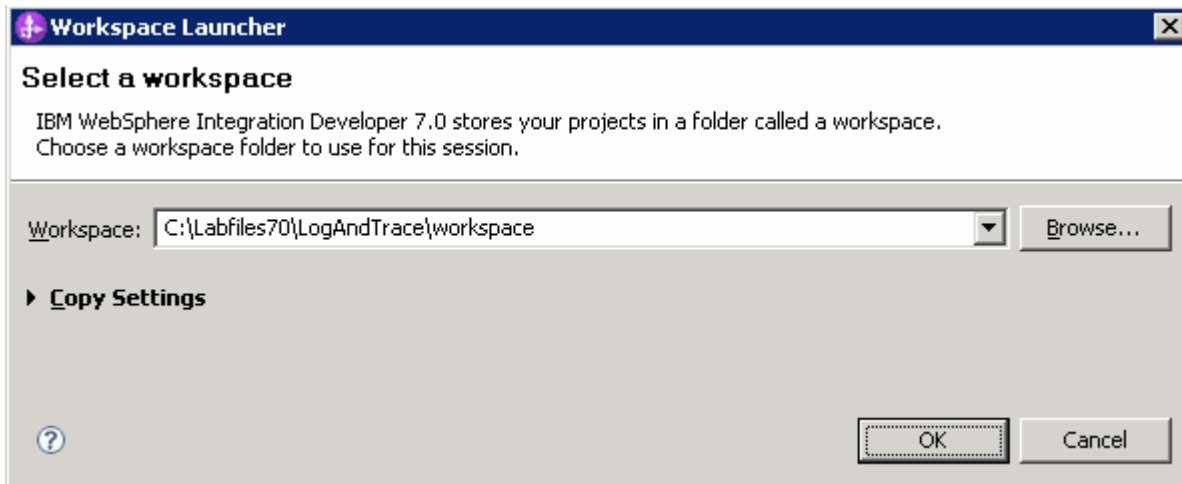
Instructions for using a remote testing environment, such as z/OS, AIX or Solaris, can be found at the end of this document, in the section [“Task: Adding remote server to WebSphere Integration Developer test environment”](#).


## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

## Part 1: Initialize the workspace and prepare for the lab

This part of the lab, you will start the WebSphere Integration Developer V7.0 with a new workspace and create required data source and database using the administrative console of WebSphere Process Server V7.0

- \_\_\_ 1. Extract the provided Labfiles70.zip to your C:\ (root) drive, if you have not already done so. This will create the necessary subdirectory structure to complete the lab, and provides you with sample text files
- \_\_\_ 2. Start the WebSphere Integration Developer V7.0 with a new workspace
  - \_\_\_ a. Select **Start > All Programs > IBM WebSphere Integration Developer > IBM WebSphere Integration Developer V7.0 > WebSphere Integration Developer V7.0**
  - \_\_\_ b. From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field



- \_\_\_ 3. Click the  button on the right to close the Welcome page and proceed with the workbench
- \_\_\_ 4. Follow the instructions of **“Define WebSphere variables”** task to define the WebSphere variables listed in the table below, which are used later in this lab:

Name	Value
LOGTRACE_EVENT	C:\Labfiles70\LogAndTrace\eventdir
LOGTRACE_ARCHIVE	C:\Labfiles70\LogAndTrace\archivedir

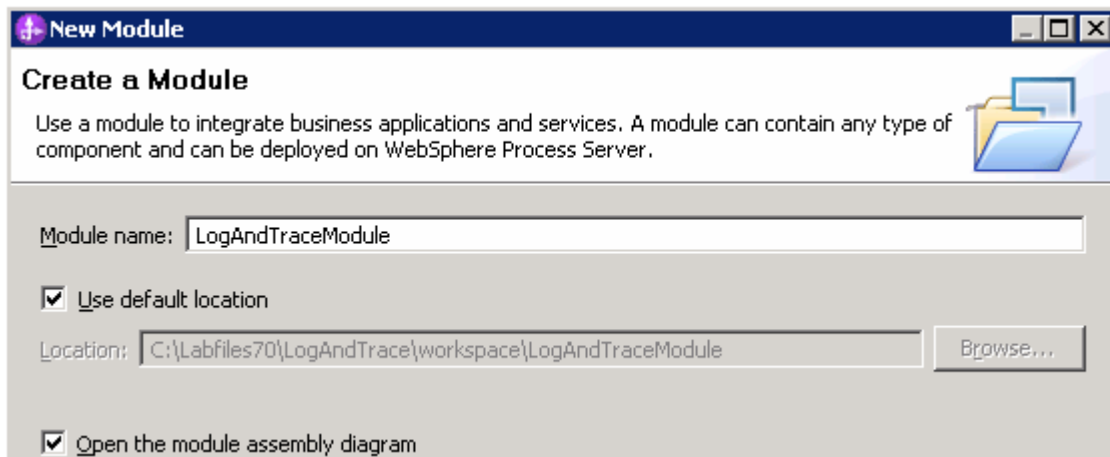
**Note:** In FlatFile adapter, you have to provide directory values (Event directory, Archive directory and so on, in activation spec and managed connection factory) while running external service wizard. Presently these values can be changed in the module (or) can be changed in deployed application from administration console. This is like hard coding the values of these properties. All local directories in ActivationSpec can be filled with **WebSphere environment variables**. This way hard coding of directory paths for these properties can be avoided. From WebSphere Process Server V6.1, you can declare these values as Environment Variables using your WebSphere Process Server administration console and specify the environment variable name in the external service wizard. So when you deploy this application, the environment variable name is replaced with actual value and used by the adapter. This is very helpful if the values of these properties need to be changed.

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

## Part 2: Configure non-pass through inbound scenario

In this part of the lab you will use the new external service feature to create and configure the function selector, data binding and other required artifacts to test the inbound pass through scenario

- \_\_\_ 1. Create the module: LogAndTraceModule
  - \_\_\_ a. From the Business Integration window, right-click and select **New > Module**
  - \_\_\_ b. From the New Module window, enter **LogAndTraceModule** for the Module Name



- \_\_\_ c. Ensure that the box next to **Open module assembly diagram** is checked and then click **Finish**

You will now see a new module, **LogAndTraceModule**, created from your Business Integration window

- \_\_\_ 2. Import required business objects
  - \_\_\_ a. Expand LogAndTraceModule (if not already expanded), right-click **Data Types** and select **Import...** from the pop-up menu
  - \_\_\_ b. From the Import window, expand **General** and select **File System** and then click **Next**
  - \_\_\_ c. Enter From directory
    - 1) Click **Browse...** next to **From directory**
    - 2) From the Import from directory window, select **<FFFILES >** and click **OK**

Now, you will see FFFiles folder added on the left side, and all the xsds and files under that folder on the right side.

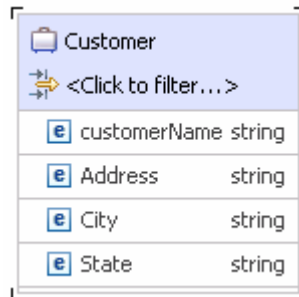
- \_\_\_ d. Select the box next to **Customer.xsd**
- \_\_\_ e. Ensure that the **LogAndTraceModule** is selected for Into folder
- \_\_\_ f. Click **Finish** from the Import window

The Business Integration window is updated with the imported business object.

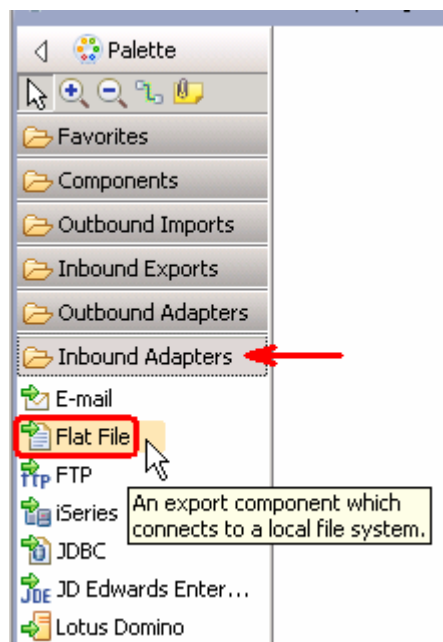
- \_\_\_ 3. Review imported business objects:

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ a. Expand **LogAndTraceModule > Data Types** and you will now see two new data types **Customer** and **Order** under it.
- \_\_\_ b. Double-click **Customer** review the fields inside the object:



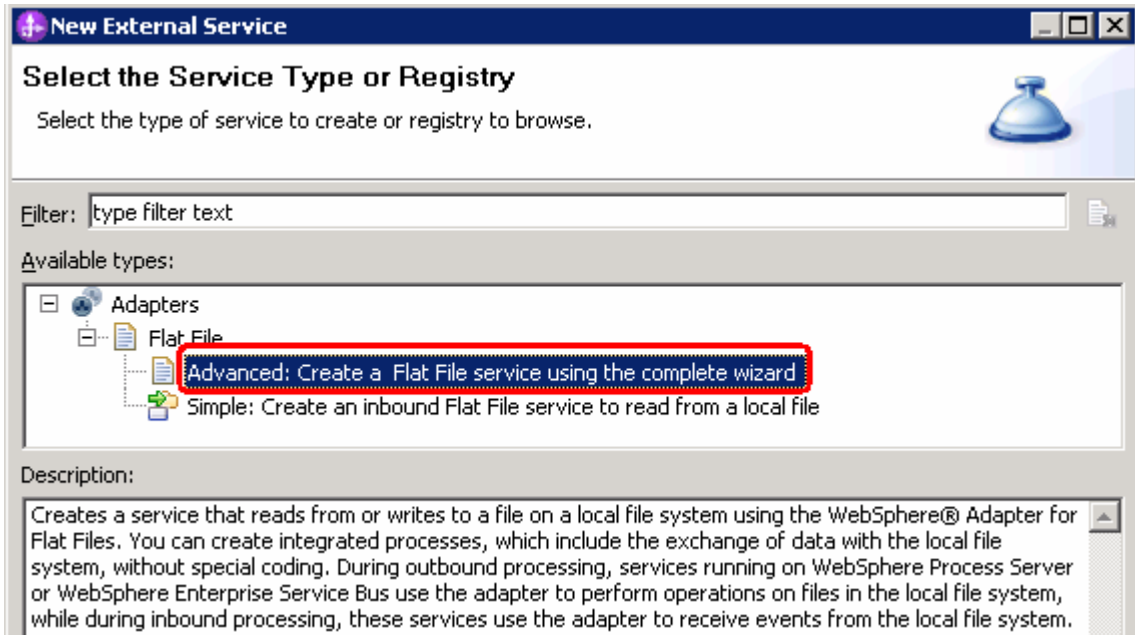
- \_\_\_ 4. After reviewing, close the Customer business object from the Assembly editor
- \_\_\_ 5. To start External Service from the Palette:
  - \_\_\_ a. From the **Palette** on the left side of Assembly Diagram, click **Inbound Adapters**:



- \_\_\_ 6. Under Inbound Adapters, click the **Flat File** and then click the empty canvas of the assembly diagram. The New Flat File Service wizard is opened

IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ 7. From the New External Service window, expand **Adapters > Flat File** and select **Advanced: Create a Flat File service using the complete wizard**



- \_\_\_ a. Click **Next**

---

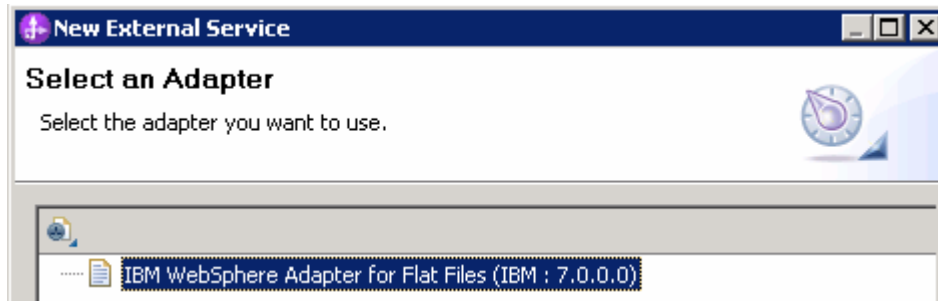
**Note:** You can also start the External Service from the **File menu** option:

From the main menu, select **File > New > External Service**. This opens an External Service wizard that helps you obtain a service which establishes connectivity with other systems.

Select **Adapters > Flat File** and click **Next**

---

- \_\_\_ 8. On the Select an Adapter screen, select **IBM WebSphere Adapter for Flat Files (IBM : 7.0.0.0)** and click **Next**



- \_\_\_ 9. Adapter Import screen:

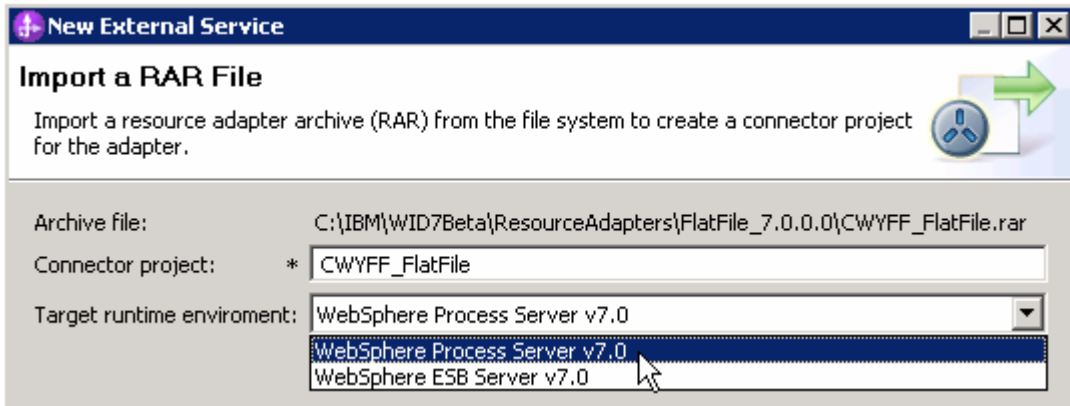
In this step, you will import a connector resource adapter archive from the file system into your WebSphere Integration Developer workspace. The adapter RAR file already exists under **<FFADAPTER\_HOME>**.

- \_\_\_ a. The default Connector file is selected which is shipped along with WebSphere Integration Developer



## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ b. Accept the default name for Connector project, **CWYFF\_FlatFile**. You can change it to any other name, but for this lab, you can leave the default name
- \_\_\_ c. For Target server, ensure that **WebSphere Process Server v7.0** is selected



- \_\_\_ d. Click **Next**

---

**Note:** The resource adapter archive file is imported and a new connector project, **CWYFF\_FlatFile**, is listed under Business Integration view.

---

---

**Note:** If you are using the **File menu** option to start the External Service wizard, you are asked to select the **Processing Direction** at this point. Select the radio button next to **Inbound** and click **Next** to proceed to the next step.

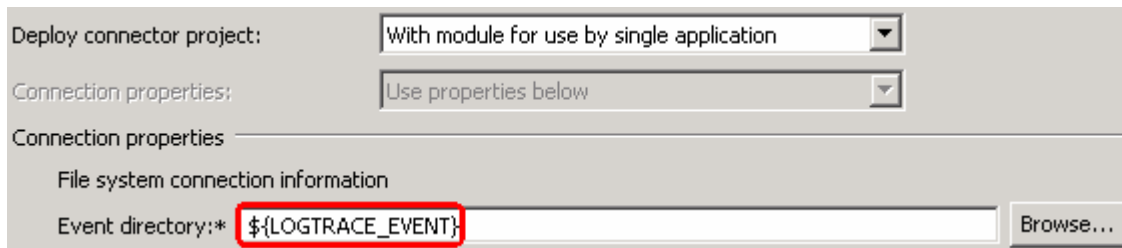
---

- \_\_\_ 10. Service Configuration Properties:
  - \_\_\_ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected
  - \_\_\_ b. Under Connection Configuration, for Event directory enter **\${LOGTRACE\_EVENT}**

---

**Note:** The WebSphere Environment Variable 'LOGTRACE\_EVENT' is already created in previous parts of this lab.

---



---

**Note:** Alternatively, you can also **Browse...** and select the event directory.

---

- \_\_\_ c. Click **Advanced >>** to see the hidden advanced properties that can be configured:

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

You can click each of the configurations and review the options available under it. For this lab, you will need only some of these properties.

\_\_ d. Event polling configuration: This has all the polling configuration details and for this lab, you can accept the defaults

\_\_ e. Event delivery configuration:

- 1) **Ensure once-only event delivery:** You should check this box only if you are using data source and table name in the Event persistence configuration (below). If this property is set to true, while using in-memory capability (explained below), the adapter will log a warning message. By default this is selected and you can accept the default selection.

\_\_ f. Event persistence configuration:

---

**Note:** This represents the JNDI name of the Data Source used by Event Persistence to get the JDBC database connection. The Data Source must be created in the WebSphere Process Server.

The Event recovery data source (JNDI) name is **not mandatory** from V6.1. Now, the adapter can use **in-memory representation** of event table to store all the necessary information. Adapter uses this feature when event database information is not configured during inbound event polling. This feature will not support the capability of handling “Ensure once-only event delivery”.

---

\_\_ g. Additional configuration:

- 1) Select the box for **Poll subdirectories in event directory**

---

**Note:** ‘**Poll subdirectories in event directory**’ is the new Boolean property added in V6.1.2. This property is used for recursive polling. Flat File adapter polls the given 'Event directory' and all the sub-directories present in the 'Event directory'.

---

The screenshot shows the 'Additional configuration' dialog box with the following settings:

- Retrieve files with pattern: \* \*.\*
- Include business object delimiter in the file content
- Retrieve files in sorted order: No sort
- File content encoding: UTF-8
- Specify the splitting function class name and the split criteria to split the file content.
  - Split file content based on size (bytes) or delimiter
  - Split function class name: com.ibm.j2ca.utls.filesplit.SplitBySize
  - Specify criteria to split file content: 0
- Poll subdirectories in event directory** (highlighted with a red box)

\_\_ h. File archiving configuration:

- 1) Archive directory: **#{LOGTRACE\_ARCHIVE}**

---

**Note:** The WebSphere Environment Variable ‘LOGTRACE\_ARCHIVE’ is already created previously in Part 1, Step 5 of this lab.

---

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

## \_\_ i. Logging and tracing:

- 1) Enter ID: **001** (enter any numeric value)

**Note:** This property identifies the adapter instance in log and trace files and for PMI events. The adapter ID is used with an adapter-specific identifier, MyAdapterRA, to form the component name used by Log and Trace Analyzer. For example, if the adapter ID property is set to 001, the component ID is FFRA001 for Flat File adapter, FTPRA001 for FTP adapter, EMARA001 for Email Adapter. For inbound processing, this property is retrieved from the resource adapter properties. For outbound processing, it is retrieved from the managed connection factory properties:

Adapter	Component ID (Filter string)
Flat File	FFRA001
FTP	FTPRA001
Email	EMA001
JDBC	JDBC001

- 2) Select the box for '**Disguise user data as "XXX" in log and trace files**'

**Note:** If you set this property to **True (selected)**, the adapter will replace user data with a string of x's when writing to log and trace files. For inbound processing, this property is retrieved from the resource adapter properties. For outbound processing, it is retrieved from the managed connection factory properties.

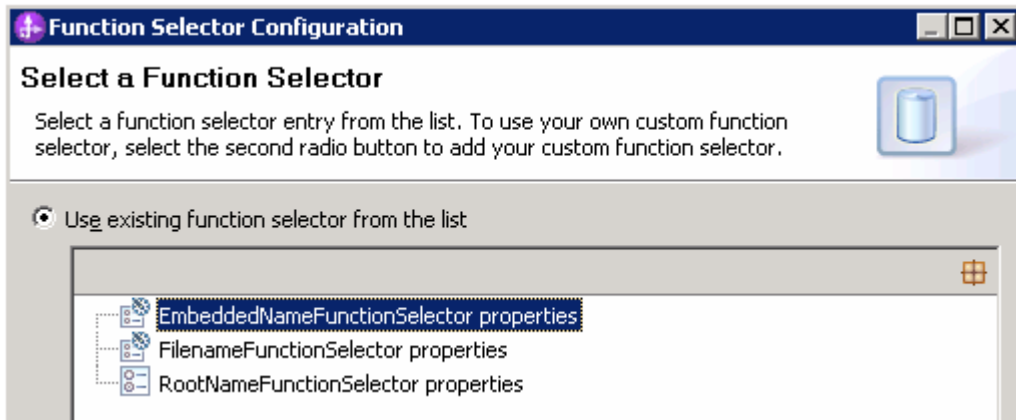
11. Under Service properties, for Function selector options, select **Use a function selector configuration** from the drop down menu

**Note:** If you select **Use default function selector 'FilenameFunctionSelector'** option for the Function selector, the adapter automatically creates a function selector with FilenameFunctionSelector as the class name. For non-pass through scenario, you need to define a function selector that uses **EmbeddedNameFunctionSelector**.

- a. Click **Select** next to Function selector. The Binding Resource Configuration window is opened

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ b. Under 'Use existing function selector from the list', select **FilenameFunctionSelector properties**



- \_\_\_ c. Click **Next**

The next screen is Function Selector Properties screen where you will define the data handler and encoding used in inbound processing.

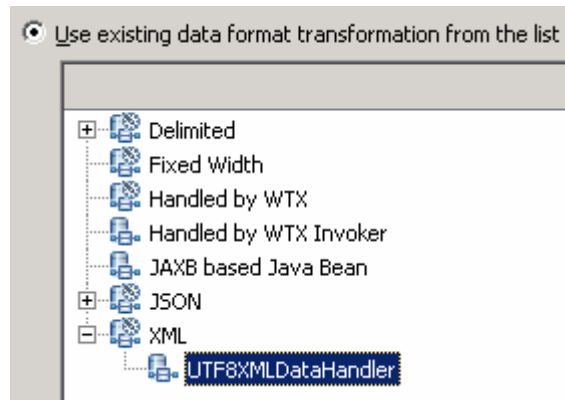
### Data handler configuration:

- \_\_\_ d. Click **Select...** next to Data handler configuration name. A Binding Resource Configuration window is opened for you to define the data handler
- \_\_\_ e. Under 'Use existing data format transformation from the list', select **XML > UTF8XMLDataHandler**

---

**Note:** UTF8XMLDataHandler listed under XML is the predefined data handler with UTF-8 as the encoding. You can also select XML and then select the encoding of your choice in the next screen to define a data handler of your choice.

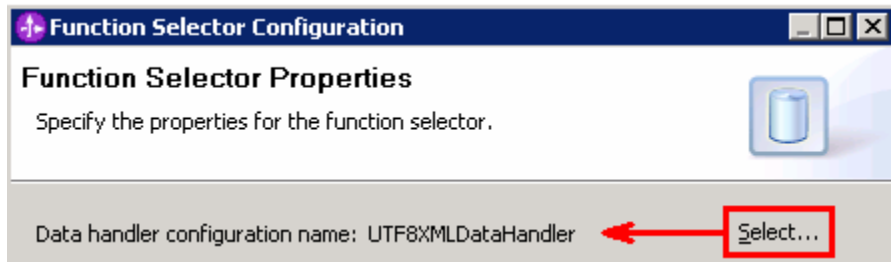
---



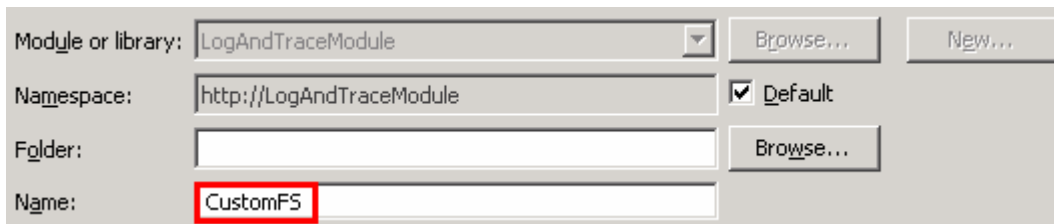
- \_\_\_ f. Click **Finish**

IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ g. From the next screen, the Data handler configuration name, **UTF8XMLDataHandler** is displayed which was selected in the previous steps

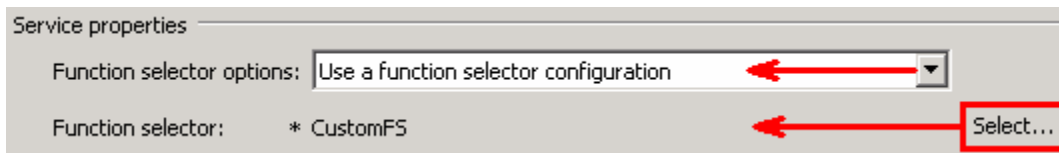


- \_\_\_ h. Click **Next**
- \_\_\_ i. From the New Function Selector Configuration screen, enter **CustomFS** for **Name**



- \_\_\_ j. Click **Finish**

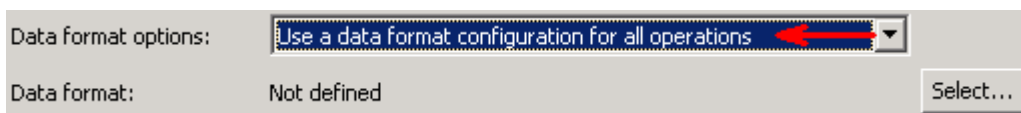
You are now done with configuring the Function Selector, CustomFS and, in that process, configured a data handler. You are back to the External Service wizard and the configured function selector, CustomFS, is displayed for Function selector configuration field:



**Data binding configuration:**

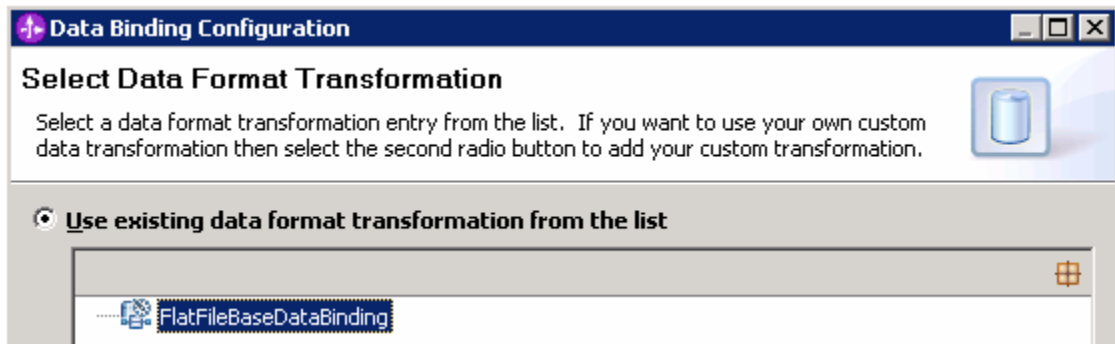
- \_\_\_ 12. You can define data binding in two places - service level (current screen of External Service wizard) or later at the method level (Operations screen of the External Service wizard). In this lab, you will define data binding at the service level (from this screen)

- \_\_\_ a. From the dropdown menu next to Data format options, select '**Use a data binding configuration for all operations**'

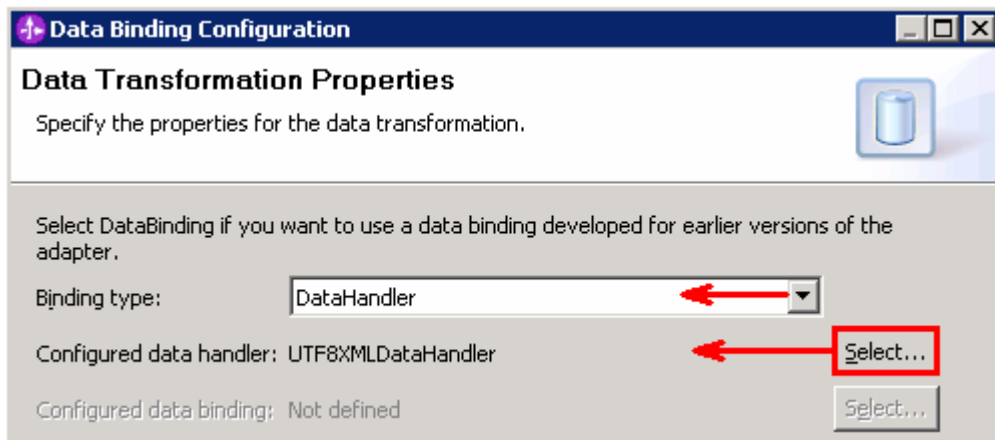


- \_\_\_ b. Click **Select...** next to **Data format**. A Binding Resource Configuration window is opened
- \_\_\_ c. Select the radio button for '**Use existing data format transformation from the list**' and then select **FlatFileBaseDataBinding**

### IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

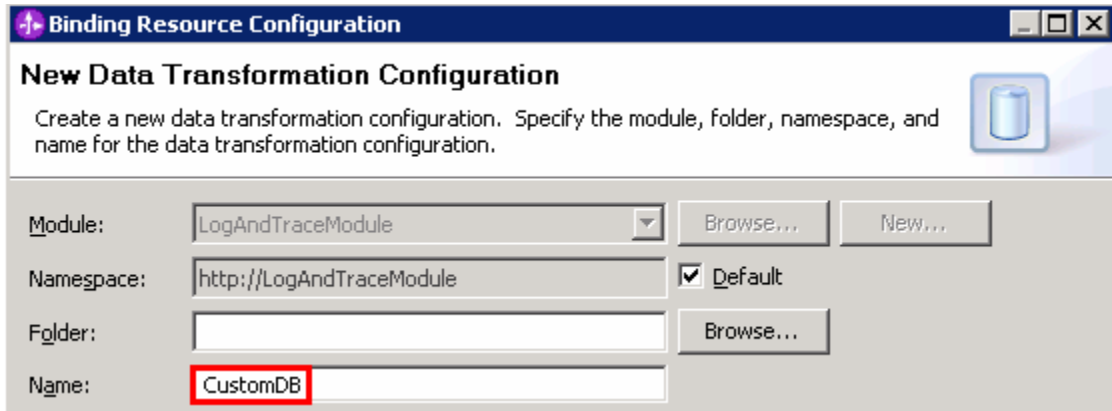


- \_\_ d. Click **Next**
- \_\_ e. Select the **UTF8XMLDataHandler** for data handler:
  - 1) Click **Select** next to 'Configured data handler'
  - 2) From the Binding Resource Configuration window, select **XML > UTF8XMLDataHandler** listed under '**Use existing data format transformation from the list**'
- \_\_ f. Click **Finish**
- \_\_ g. Back to Data Transformation Properties and the selected data handler '**UTF8XMLDataHandler**' is displayed here:



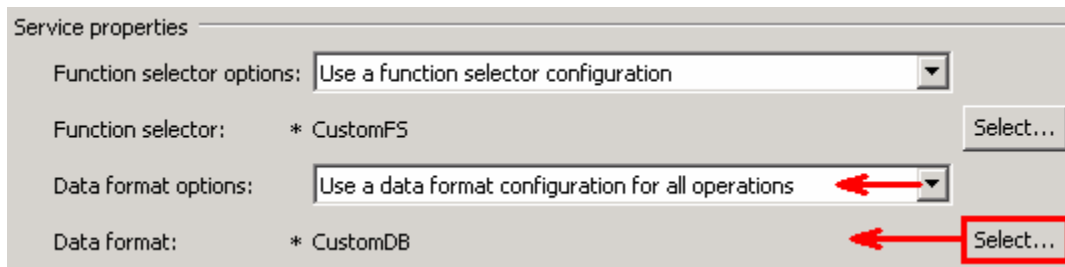
- \_\_ h. Click **Next**
- \_\_ i. Note that the selected module is **LogAndTraceModule**
  - 1) For the **Name**, enter **CustomDB**

IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE



2) Click **Finish**

\_\_\_ j. Now the **CustomDB** should be displayed for Data binding configuration

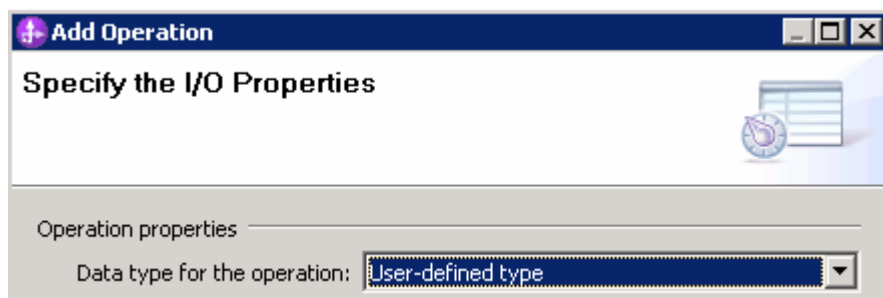


\_\_\_ 13. Check the box next to **Change logging properties for wizard** to view the output location of the log file and the logging level and click **Next**

**Define emitCustomerFile operation:**

\_\_\_ 14. From the Operations screen, click **Add....** Add Operation window is opened

\_\_\_ a. Select **User-defined type** for the Data type and click **Next**



You are now back to Operation window and because you chose the User defined data type, the Input type is **blank** and because you have selected Output required box, the Output type is **CreateResponse**

\_\_\_ b. For Operation name, enter **emitCustomerFile**

\_\_\_ c. Define Input type:

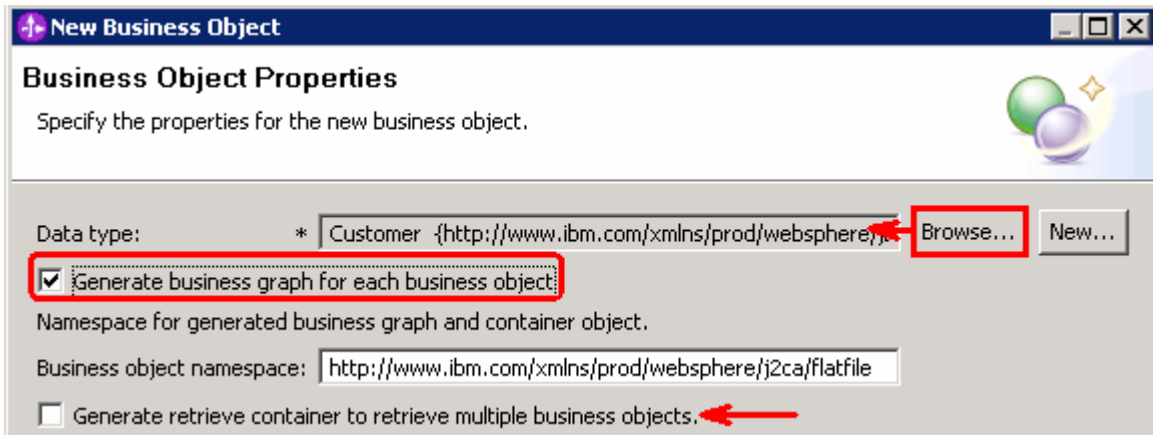
1) Click **New...** next to **Input type** to open a New Business Object window

### IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- 2) From this window, ensure that the Module selected is **LogAndTraceModule** and click **Next**
- 3) Click **Browse...** next to Data type
- 4) From the Data Type Selection window, select **Customer** under Matching data types:

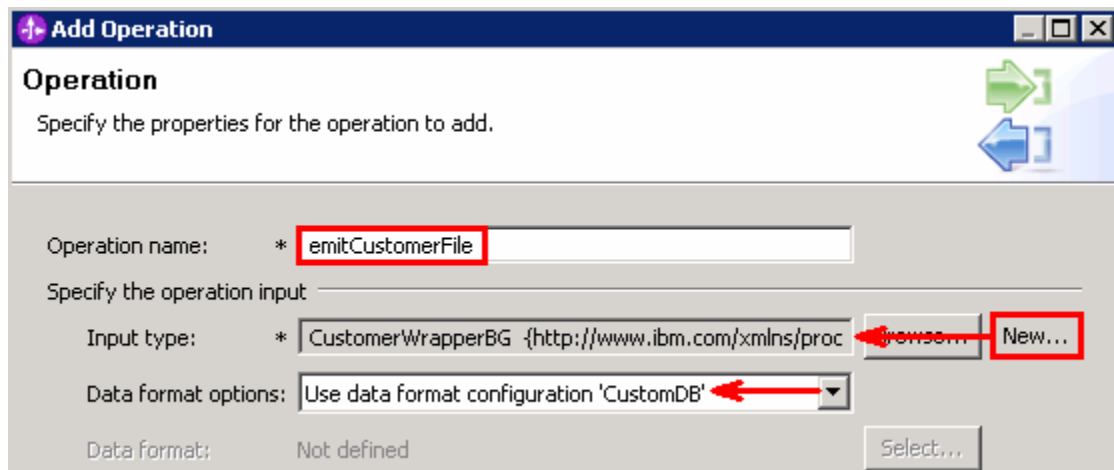


- 5) Click **OK**
- 6) From the wizard, **check** the box next to **Generate business graph for each business object**
- 7) **Do not** check the box for '**Generate retrieve container to retrieve multiple business objects**'. This is used for outbound retrieve operation.



- 8) Click **Finish**

- \_\_\_ d. The Input type in Add Operation window is populated with CustomerWrapperBG, because you have selected to have business graph (BG) generated:
- \_\_\_ e. For **Data format options**, accept the default selection **Use data format configuration 'CustomDB'** from the dropdown list

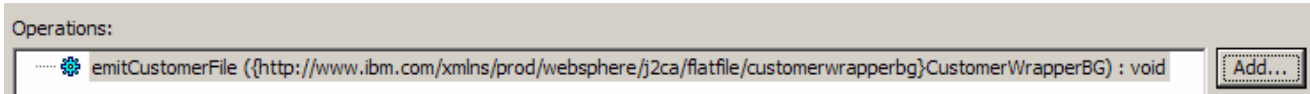




### IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

\_\_ f. Click **Finish**

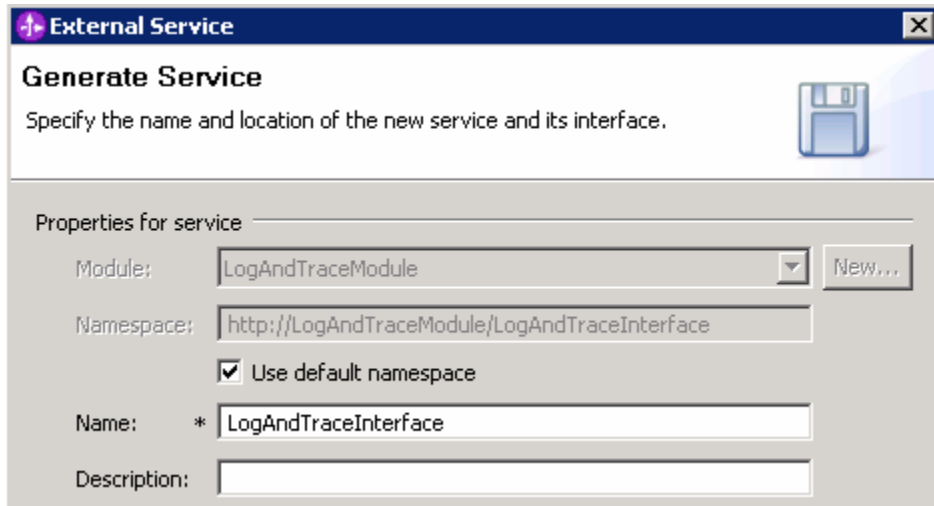
\_\_ g. The defined operation, emitCustomerFile, is populated in the Operations list



\_\_ h. Click **Next**

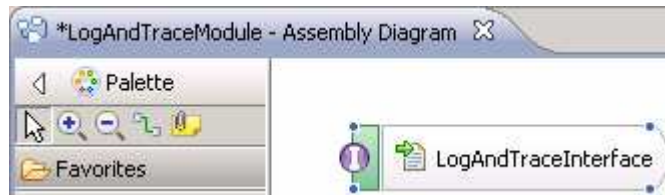
\_\_\_ 15. From the Generate Artifacts screen, enter these:

\_\_ a. For Name, enter **LogAndTraceInterface**



\_\_ b. Click **Finish**

\_\_\_ 16. You will now see a new export component, **LogAndTraceInterface** in the assembly diagram of FFCustomInboundModule



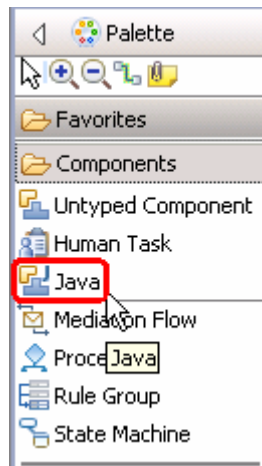
\_\_ a. Save (**Ctrl+S**) your changes and close the assembly diagram


## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

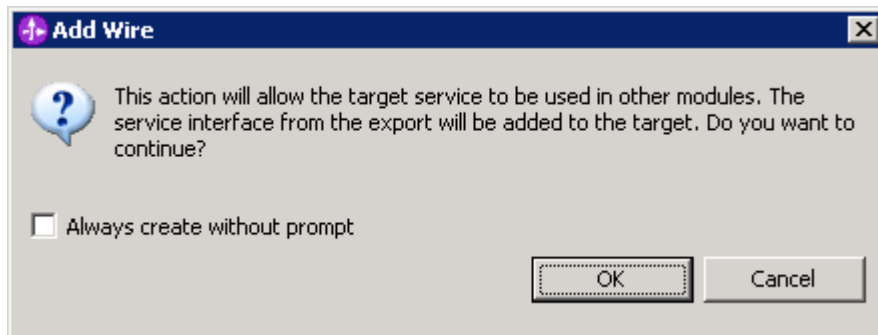
## 2.1. Add Java component

In this part of the lab, you will add a Java component to the module and then wire the export to the added component. Then you will add the custom Java code to the added module.

- \_\_\_ 1. Open the assembly diagram for LogAndTraceModule, if it is already not open
  - \_\_\_ a. From the business integration view, expand **LogAndTraceModule** and double click **Assembly diagram**
- \_\_\_ 2. Drop a Java component to onto the assembly diagram
  - \_\_\_ a. From the Palette, click Components to expand it



- \_\_\_ b. Click **Java** and then click the empty space of LogAndTraceModule assembly diagram. This will place a new component, Component1 on the assembly diagram.
- \_\_\_ 3. Wire the LogAndTraceInterface to the Component1
  - \_\_\_ a. Select the **wire** (  ) icon from the Palette
  - \_\_\_ b. Click **LogAndTraceInterface** and then click **Component1** to wire them together
  - \_\_\_ c. Select **OK** for the Add Wire pop-up window:



## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

Your assembly diagram for LogAndTraceModule will look like this:



\_\_ d. From the top of the Palette, click the **Selection Tool** icon (  ) to get back to the normal cursor mode

\_\_\_ 4. Generate Java Implementation

\_\_ a. Right-click **Component1** and select **Generate Implementation** from the pop-up menu

\_\_ b. On the **Generate Implementation** panel, select default package, and click **OK**

\_\_ c. **Component1Impl.java** is opened in Assembly editor. Scroll down to the method **emitCustomerFile(DataObject emitCustomerFileInput)** that needs to be implemented and add this code under that method:

```
System.out.println("*****ENDPOINT emitCustomer*****");
DataObject wrapper =
emitCustomerFileInput.getDataObject("CustomerWrapper");
String filename = wrapper.getString("fileName");
System.out.println("FILENAME : "+filename);
DataObject customer = wrapper.getDataObject("Content");
String name = customer.getString("customerName");
System.out.println("NAME-----> "+name);
String address = customer.getString("Address");
System.out.println("ADDRESS--> "+address);
String city = customer.getString("City");
System.out.println("CITY-----> "+city);
String state = customer.getString("State");
System.out.println("STATE-----> "+state);
```

---

**Note:** The code is also available at <FFFILES>\CustomerJavaCode.txt for your convenience

---

\_\_ d. Save (**Ctrl + S**) and close Component1Impl.java

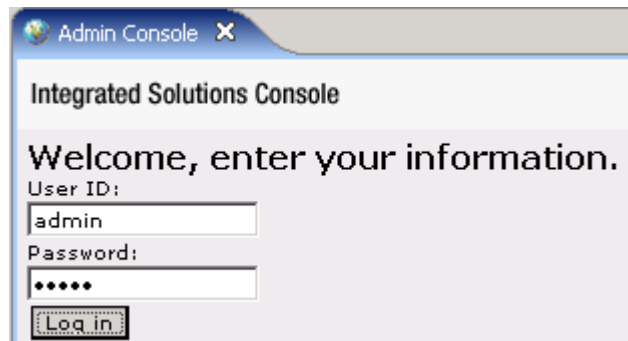
\_\_\_ 5. Save (**Ctrl + S**) and close Assembly diagram: LogAndTraceModule

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

## Part 3: Deploy the application and enable tracing

In this part of the lab, you will use the WebSphere Process Server Test Environment to test the SCA application Inbound processing for the pass through scenario.

- \_\_\_ 1. Start the server (if not already started)
  - \_\_\_ a. From the Servers view, right-click the **WebSphere Process Server v7.0** and select **Start**
  - \_\_\_ b. Wait until the server status turns to **Started**
- \_\_\_ 2. Add the project to the WebSphere Test Environment server
  - \_\_\_ a. Right-click **WebSphere Process Server v7.0** under the Servers view and select **Add and remove projects...** from the pop-up menu
  - \_\_\_ b. From the Add and Remove Projects window, select **LogAndTraceModuleApp** under Available projects panel and click **Add >**
  - \_\_\_ c. You will now see the **LogAndTraceModuleApp** added to the **Configured projects**
  - \_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server will start in Debug mode if it is not already started before
- \_\_\_ 3. Enable tracing
  - \_\_\_ a. Right-click WebSphere Process Server v7.0 and select **Administration > Run administrative console**
  - \_\_\_ b. From the 'Admin Console' window, enter Administrative User ID and Password and then click **Log in**



- \_\_\_ c. Expand **Troubleshooting** and click **Logs and Trace**

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

\_\_\_ d. From the Logging and Tracing screen, click **server1**

**Logging and Tracing**

Use this page to specify how the server handles log records. You can select an application server to enable or disable a system log for that server, specify where log data is stored, and choose a format for log content. You can also specify a log detail level for components and groups of components.

⊕ Preferences

Server	Node	Host Name	Version	Type
You can administer the following resources:				
<a href="#">server1</a>	qnode	AIMCP103.austin.ibm.com	ND 7.0.0.7 Process Choreographer 7.0.0.0 SCA FEP 1.0.1.0 WPS 7.0.0.0 XML FEP 1.0.0.0	servers
Total 1				

\_\_\_ e. Click **Change Log Detail Levels** under General Properties

**Logging and Tracing > server1**

Use this page to select a system log to configure, or to specify a log detail level for components and groups of components. Use log levels to control which events are processed by Java logging.

**General Properties**

- [Diagnostic Trace](#)
- [JVM Logs](#)
- [Process Logs](#)
- [IBM Service Logs](#)
- [Change Log Detail Levels](#)

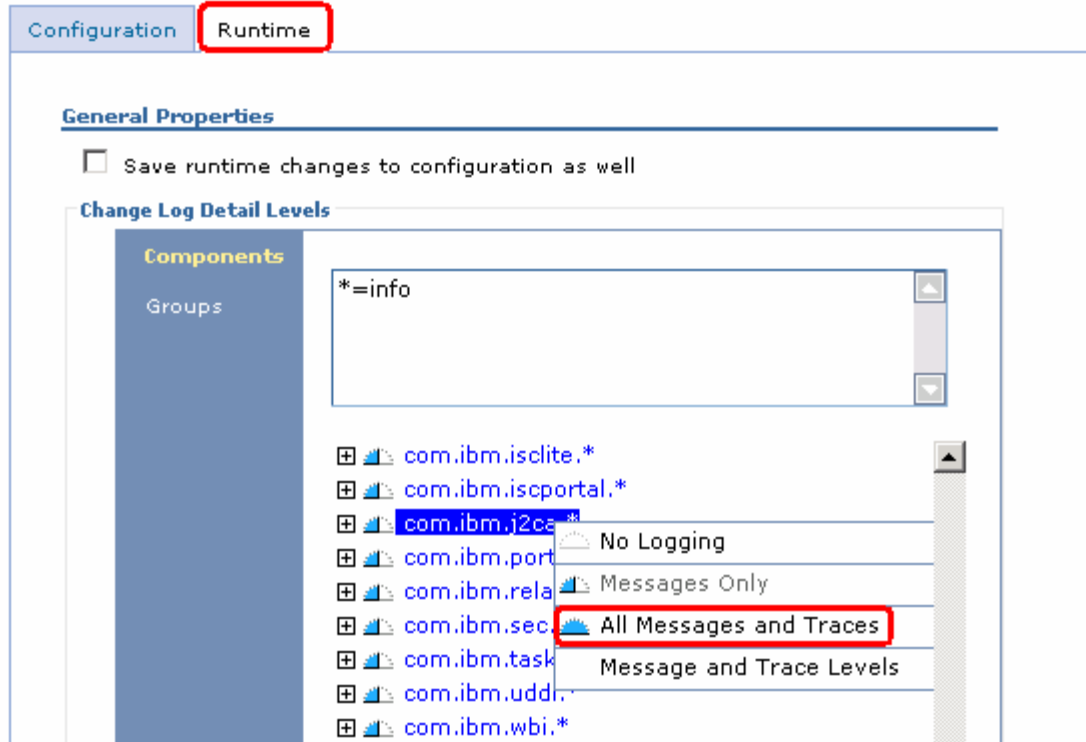
\_\_\_ f. Select the **Runtime** tab

IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

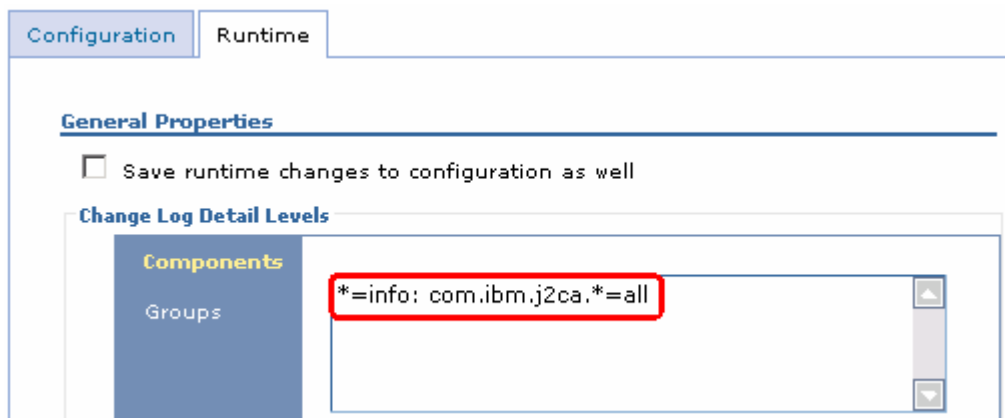
- \_\_ g. Scroll down and click **com.ibm.j2ca.\*** under Change Log Detail Levels and then select **All Messages and Traces**

[Logging and Tracing](#) > [server1](#) > **Change Log Detail Levels**

Use log levels to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative; a level near the top of the list includes all the subsequent levels.



- \_\_ h. You should see the below statement:



- \_\_ i. Click **OK** at the bottom
- \_\_ j. Click **Save** at the top to save all your changes
- \_\_ k. **Log Out** and close the administrative console window

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

**Part 4: Test the application and review confidential messages**

\_\_\_ 1. Provide the input file:

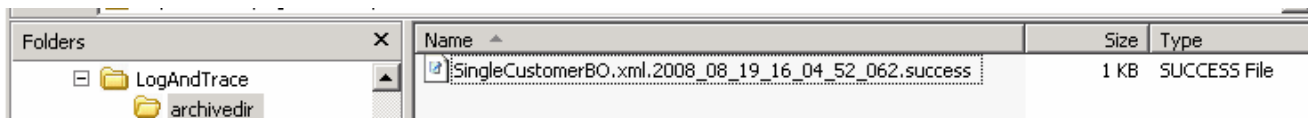
**Note:** For your convenience, a sample test file, **SingleCustomerBO.xml**, which contains one customer business object, is placed in **<FFFILES>**.

\_\_\_ a. Copy the **sample.txt** file from **<FFFILES>** to **<EVENT\_DIR>/subdir**. The adapter will poll the copied file from the sub directory under event directory and will transfer it to the archive directory

\_\_\_ b. Because you have placed a file that contains customer business object, it will pass through the **emitCustomerFile** method and you should see this message in the console (or SystemOut.log) or from the Server Logs view:

```
SystemOut      O *****ENDPOINT emitCustomer*****
SystemOut      O FILENAME : SingleCustomerBO.xml
SystemOut      O NAME-----> IBM
SystemOut      O ADDRESS--> 11501 Burnet Rd
SystemOut      O CITY-----> Austin
SystemOut      O STATE----> TX
```

\_\_\_ c. To verify your test results, check the **<ARCHIVE\_DIR>** subdirectory which should contain an archive of the event file, with the same file name appended with year, month, date, system time, and success



\_\_\_ 2. Review trace file for confidential messages:

- **Flat File:** When confidential tracing is enabled, BO data for both inbound and outbound is replaced by XXXXX. For outbound (create, append, and overwrite) the BO is not masked since the BO content is in the request and at the databinding, it is not possible to differentiate if confidential tracing was enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did not originate inside the EIS. This includes, but is not limited to: Transaction IDs, Event IDs, BO schemas, and Call sequences.
- **FTP:** When confidential tracing is enabled: BO data for both inbound and outbound is replaced by XXXXX. Username is replaced by XXXXX. Password will always be written to the trace file as \*\*\*\*\* irrespective of whether confidential tracing is enabled or not. FTP URL is traced to the trace file. This is part of each trace message (for example: ftp://localhost/eventDir). This is required because when you are using multiple Activation Spec (AS) or MCFs with each AS polling one dir on the same FTP server, it is easier to identify to which AS the traces belong to. For outbound (create, append, and overwrite) the BO is not masked since the BO content is in the request and at the databinding it is not possible to differentiate if confidential tracing was enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did not originate inside the EIS. This includes, but is not limited to: Transaction IDs Event IDs BO Schemas Call sequences
- **Email:** When confidential tracing is enabled: BO data for both inbound and outbound is replaced by XXXXX. Username is replaced by XXXXX. Password will always be written to the trace file as \*\*\*\*\* irrespective of whether confidential tracing is enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

not originate inside the EIS. This includes, but is not limited to: Transaction IDs Event IDs BO Schemas Call sequences

- \_\_\_ a. Open your windows explorer and open <WPS\_HOME>\logs\server1\trace.log using note pad
- \_\_\_ b. Note that the messages have 'FFRA001' in their description and this is because you specified the Adapter ID as 001 while configuring the adapter Also, you should see 'XXXX' replaced for all the confidential data:

```
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
traceActivationSpecProperties EP_UserName = XXXX
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
traceActivationSpecProperties EP_Password = XXXX
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
traceActivationSpecProperties EP_TableName = XXXX
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
traceActivationSpecProperties EP_SchemaName = XXXX
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
traceActivationSpecProperties EP_DataSource_JNDIName = XXXX
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001 < com.ibm.j2ca.flatfile.FlatFileResourceAdapter

[10/3/08 11:52:05:812 CDT] 000000e0 FFRA001 3 com.ibm.j2ca.flatfile.util.FlatFileUtil
getTotalBosFromEventId Total number of BOS is 1
[10/3/08 11:52:05:812 CDT] 000000e0 FFRA001 3
com.ibm.j2ca.flatfile.FlatFileEventstorewithxid getRecordForEvent BO Content : XXXX
[10/3/08 11:52:05:812 CDT] 000000e0 FFRA001 3
com.ibm.j2ca.flatfile.FlatFileEventstorewithxid getRecordForEvent DirectoryPath set in the
record object is C:\Labfiles62\LogAndTrace\eventdir
```

### \_\_\_ 3. Restore the Sever Configuration

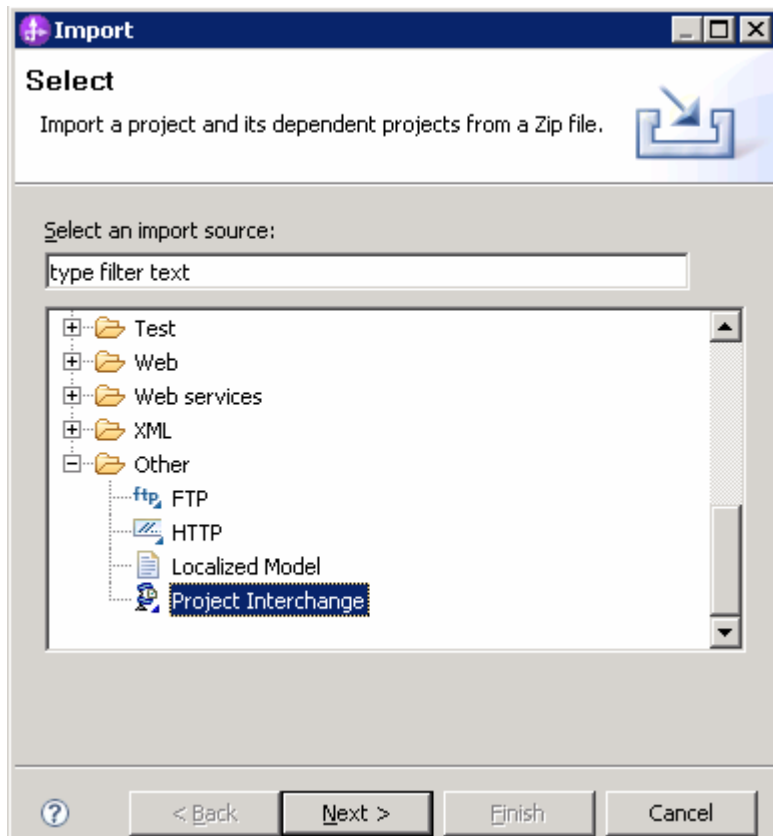
- \_\_\_ a. Right-click **WebSphere Process Server v7.0** under the Servers view and select **Add and remove projects...** from the pop-up menu
- \_\_\_ b. Select **LogAndTraceModuleApp** under Configured projects and click **< Remove**
- \_\_\_ c. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished



## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

### Solution instructions

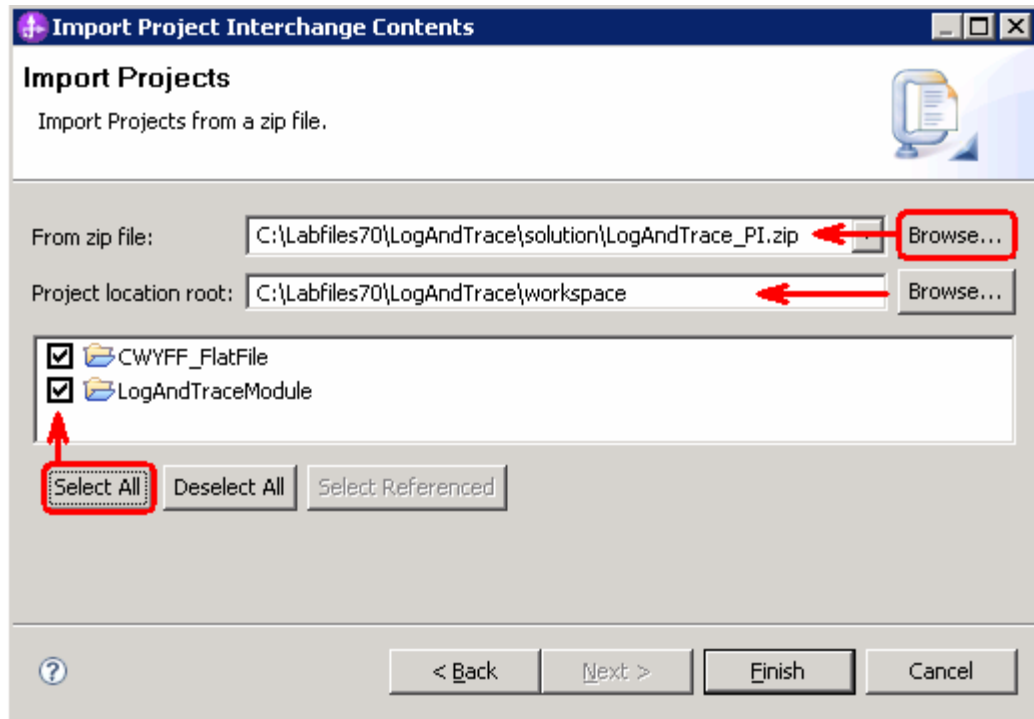
- \_\_\_ 1. Start WebSphere Integration Developer V7.0 with a new workspace
  - \_\_\_ a. Follow the instructions outlined in **Part 1** of this exercise
- \_\_\_ 2. Import the solution Project Interchange
  - \_\_\_ a. Import the project interchange file **LogAndTrace\_Pi.zip** from **<LAB\_FILES>\LogAndTrace\solution** directory
  - \_\_\_ b. Select **File → Import** from the menu
  - \_\_\_ c. Select **Other → Project Interchange** in the **Import** dialog and click **Next**



- \_\_\_ d. For the **From zip file**, click on the **Browse** button and select the **LogAndTrace\_Pi.zip** in the **<LAB\_FILES>\LogAndTrace\solution** directory
- \_\_\_ e. Enter **<LAB\_FILES>\LogAndTrace\workspace** for the **Project location root**

### IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ f. Click the **Select All** button. This will select all modules: **CWYFF\_FlatFile**, and **LogAndTraceModule**



- \_\_\_ g. Click **Finish**

- \_\_\_ 3. Test the scenario: Continue with **Part 3** and **Part 4** of this

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

### What you did in this exercise

In this lab, you started with importing the Flat File Adapter RAR file into your WebSphere Integration Developer new workspace. Next, you made use of the External Service wizard available in WebSphere Integration Developer to specify Activation Spec Properties, define Data binding, Data handler, and Function selector which, after deploying onto the server, will generate Business Objects and other artifacts. Then you continued to complete building the application by adding a Java component and implementation.

Next, you deployed the adapter application onto WebSphere Process Server and enable tracing in WebSphere Process Server administrative console and tested the application for sample input file containing single customer business object.

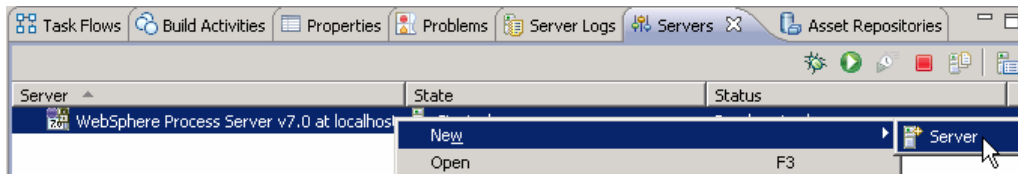
You loaded the trace file into the Server Logs view, and reviewed the messages matching Adapter ID. Finally, you reviewed the trace file contents for the 'confidential' messages replaced with 'XXX'.

## IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

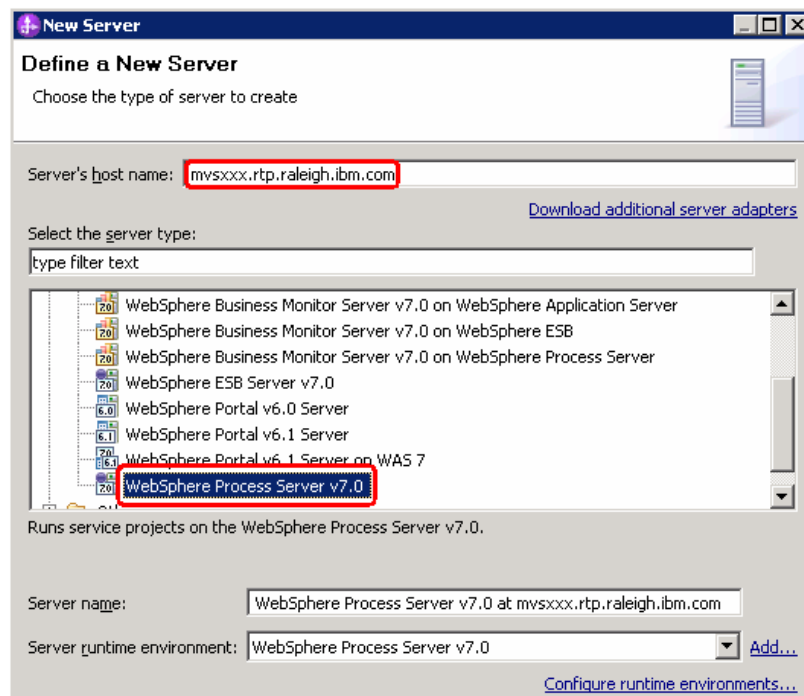
## Task: Adding remote server to WebSphere Integration Developer test environment

This task describes how to add a remote server to the WebSphere Integration Developer Test environment. This example uses a z/OS machine.

- \_\_\_ 1. Define a new remote server to WebSphere Integration Developer.
  - \_\_\_ a. Right click the background of the Servers view to access the pop-up menu.
  - \_\_\_ b. Select **New → Server**.



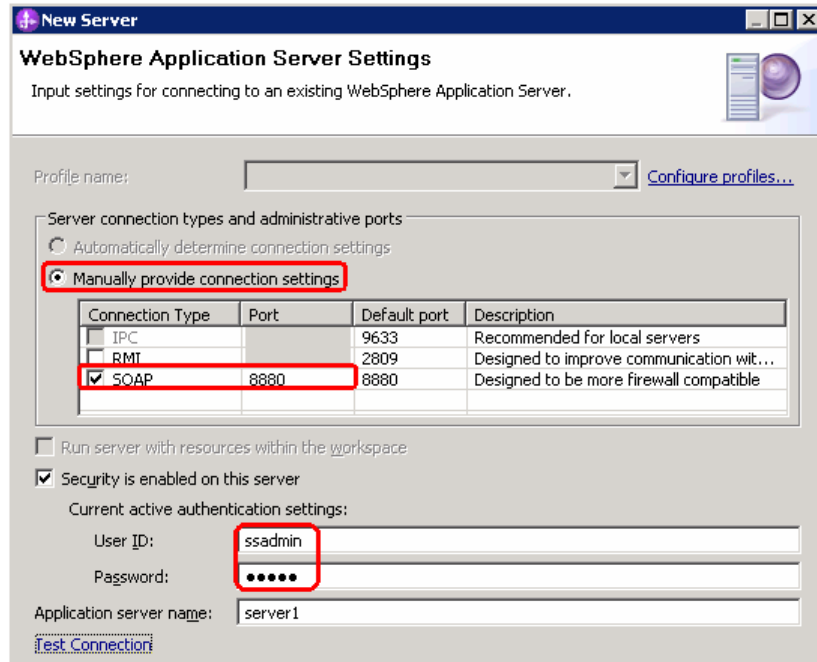
- \_\_\_ c. In the New Server dialog, specify the remote server's host name, **<HOSTNAME>**.
- \_\_\_ d. Ensure that the appropriate server type, **'WebSphere Process Server v7.0'** or **'WebSphere ESB Server v7.0'**, is highlighted in the server type list



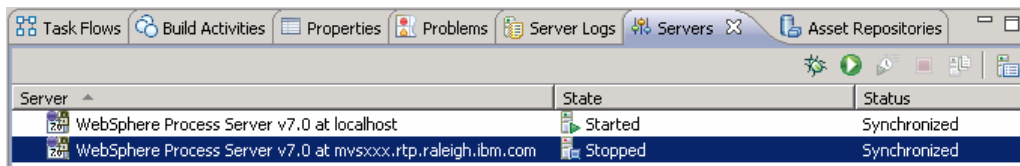
- \_\_\_ e. Click **Next**.

IBM WEBSHERE ADAPTER 7.0 – LAB EXERCISE

- \_\_\_ f. On the WebSphere Server Settings page, leave the radio button for **Manually provide connection settings** selected, and select the box for SOAP
- \_\_\_ g. Enter the correct setting (<SOAP\_PORT>) for **Port** column
- \_\_\_ h. If security is enabled on your server, select the box for ‘**Security is enabled on this server**’ and enter <USERID> for the user ID and <PASSWORD> for the password.



- \_\_\_ i. Click **Finish**.
- \_\_\_ j. The new server should be seen in the Server view.



- \_\_\_ 2. Start the remote server if it is not already started. WebSphere Integration Developer does not support starting remote servers from the Server view.
- \_\_\_ a. From a command prompt, telnet to the remote system if needed:

**'telnet <HOSTNAME> <TELNET\_PORT>'**

User ID : <USERID>

Password : <PASSWORD>

## IBM WEBSPHERE ADAPTER 7.0 – LAB EXERCISE

\_\_ 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 sssr01 open for e-business; process id is 0000012000000002
```