

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Flat File adapter – Processing COBOL copybook files lab

What this exercise is about	1
Lab requirements	1
What you should be able to do	2
Introduction	2
Exercise instructions	2
Part 1: Initialize the workspace and prepare for the lab.....	5
Part 2: Inbound scenario	6
2.1. Prepare for the inbound scenario.....	7
2.2. Configure inbound using the external service wizard	12
2.3. Add Java component	17
2.4. Test inbound scenario.....	19
Part 3: Outbound scenario	21
3.1. Prepare for the outbound scenario	22
3.2. Configure outbound using external service wizard	24
3.3. Test outbound scenario	30
What you did in this exercise	38
Task: Adding remote server to WebSphere Integration Developer test environment	39

What this exercise is about

The objective of this lab is to provide you with an understanding of how to use the 'external data' wizard in WebSphere Integration Developer to generate business object definitions from a COBOL program source file. Then, the lab guides you through the configuration of inbound and outbound processing using these business object definitions to process COBOL copybook files.

Lab requirements

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

- WebSphere Integration Developer V6.2 installed and updated with latest fixes
- WebSphere Process Server V6.2 Test Environment installed and updated latest fixes
- Extract Labfiles62.zip to your C:\ (your root) drive

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

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 Data wizard to generate business object and wrapper definitions from a COBOL program source file
 - Use External Service wizard to configure Activation Spec Properties, Resource Adapter Properties to generate Business Objects and other artifacts
 - Deploy the adapter application onto WebSphere Process Server
 - Test the deployed application using WebSphere Process Server test environment
 - Restore the server configuration
-

Introduction

Use the external data wizard to generate a business object definition for a COBOL program source file. After you have generated the business object definition, you can optionally rerun the external data wizard to generate a wrapper business object definition from the generated business object.

You will use external service wizard to configure Flat File adapter to convert COBOL copybook files to business objects during the inbound processing and convert business objects to COBOL copybook files during outbound processing.

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

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

Reference variable	Windows location	AIX® or UNIX® location
<WID_HOME>	C:\Program Files\IBM\WID62	
<WPS_HOME>	C:\<WID_HOME>\runtimes\bi_v62	
<FFADAPTER_HOME>	<WID_HOME>\ResourceAdapters\FlatFile_6.2.0.0\deploy	
<LAB_FILES>	C:\Labfiles62	/tmp/Labfiles62
<WORKSPACE>	<LAB_FILES>\FlatFileCobol\workspace	
<EVENT_DIR>	<LAB_FILES>\FlatFileCobol\eventdir	
<ARCHIVE_DIR>	<LAB_FILES>\FlatFileCobol\archivedir	
<OUT_DIR>	<LAB_FILES>\FlatFileCobol\outdir	
<FFFFILES>	<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:\Labfiles62\ with C:/Labfiles62/

IBM WEBSHERE ADAPTER 6.2 – 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 following table 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	

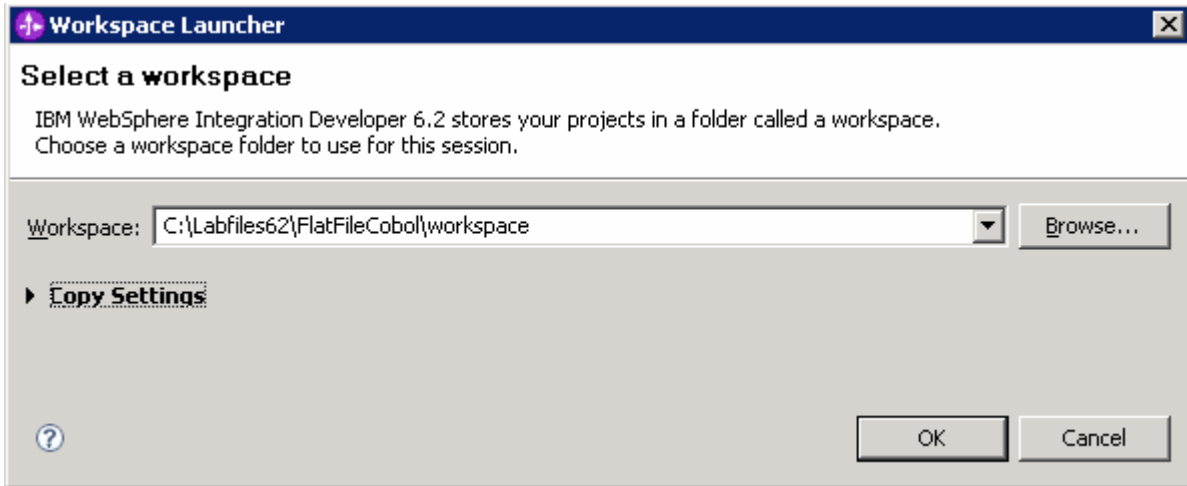
Instructions for using a remote testing environment, such as z/OS, AIX or Solaris, can be found at the end of this document, in the section [“Task: Adding remote server to WebSphere Integration Developer test environment”](#).


IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Part 1: Initialize the workspace and prepare for the lab

This part of the lab, you will extract the required lab files and start the WebSphere Integration Developer V6.2 with a new workspace and then define required WebSphere Environment Variables using the administrative console of WebSphere Process Server V6.2

- ___ 1. Extract the provided Labfiles62.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 V6.2 with a new workspace
 - ___ a. Select **Start > All Programs > IBM WebSphere Integration Developer > IBM WebSphere Integration Developer V6.2 > WebSphere Integration Developer V6.2**
 - ___ b. From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field



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

Name	Value
FF_EVENT	C:\Labfiles62\FlatFileCobol\eventdir
FF_ARCHIVE	C:\Labfiles62\FlatFileCobol\archivedir
FF_OUT	C:\Labfiles62\FlatFileCobol\outdir

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Part 2: Inbound scenario

In this part, you will import the Flat File adapter RAR into the new workspace and use the external data wizard to generate a business object definition for a COBOL program source file. After you have generated the business object definition, you can optionally rerun the external data wizard to generate a wrapper business object definition from the generated business object.

You will make use of the new 'COBOL, C, or PL/I data binding' introduced in V6.2, while running the External service wizard to generate the required artifacts to convert COBOL copybook files to business objects during inbound processing.

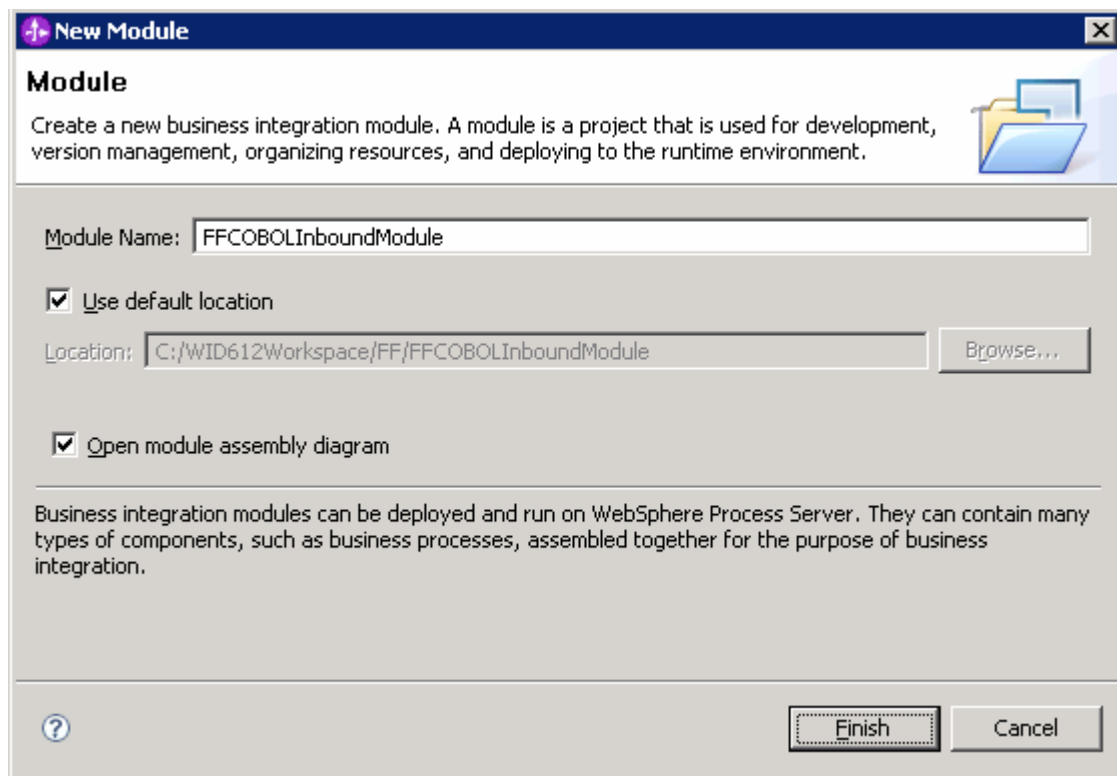
IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

2.1. Prepare for the inbound scenario

In this part of the lab you will create a module, use the external data wizard to generate a business object definition for a COBOL program source file. After you have generated the business object definition, you can optionally rerun the external data wizard to generate a wrapper business object definition from the generated business object.

1. Create FFCOBOLInboundModule

- ___ a. From the Business Integration window, right-click and select **New > Module**
- ___ b. From the New Module window, enter **FFCOBOLInboundModule** for the Module Name
- ___ c. Ensure that the box next to **Open module assembly diagram** is checked



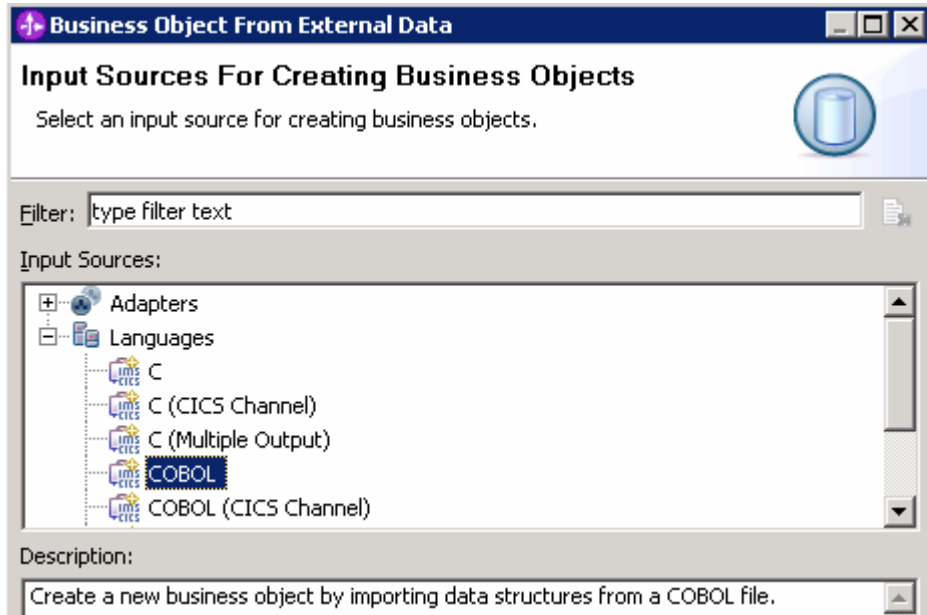
- ___ d. click **Finish**
- ___ e. You will now see a new module, FFCOBOLInboundModule, created in your Business Integration window

2. Generate the business object definition for the COBOL program source file

- ___ a. In the Business Integration section of the window, right-click the module, FFCOBOLInboundModule, and select **New > Business Object from External Data**

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

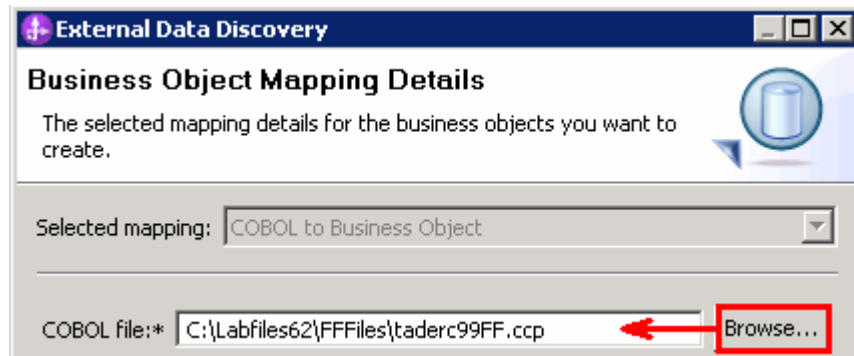
__ b. From the Business Object from External Data window, select **Languages > COBOL**



__ c. Click **Next**

__ d. In the Business Object Mapping Details window,

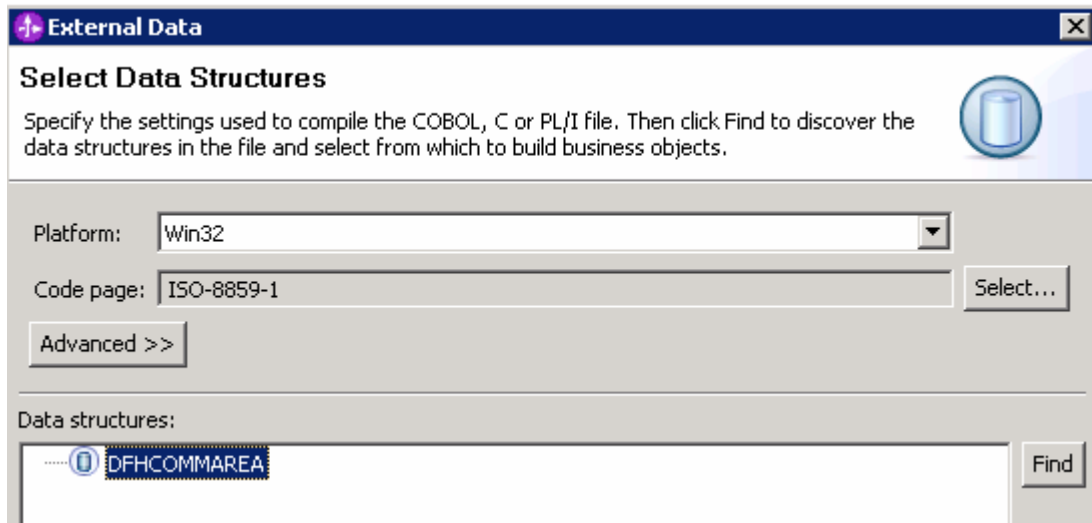
- 1) Note that the **Selected mapping** value is **COBOL to Business Object**
- 2) For COBOL file, click **Browse...** and select the **traderc99FF.ccp** file located in **<FFFILES>**



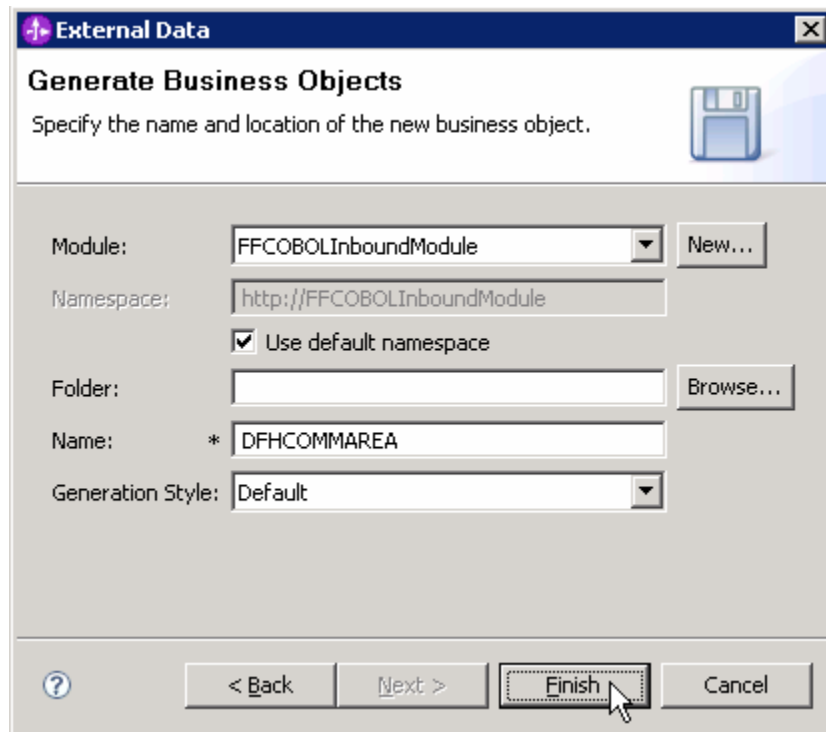
3) Click **Next**

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

- ___ e. In the Select Data Structures window, the new business object, called **DFHCOMMAREA**, is displayed

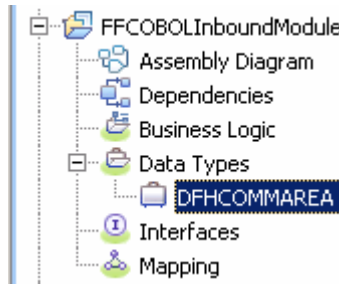


- ___ f. Select **DFHCOMMAREA** and click **Next**
- ___ g. In the Generate Business Objects screen, ensure that the selected Module is **FFCOBOLInboundModule** and click **Finish**

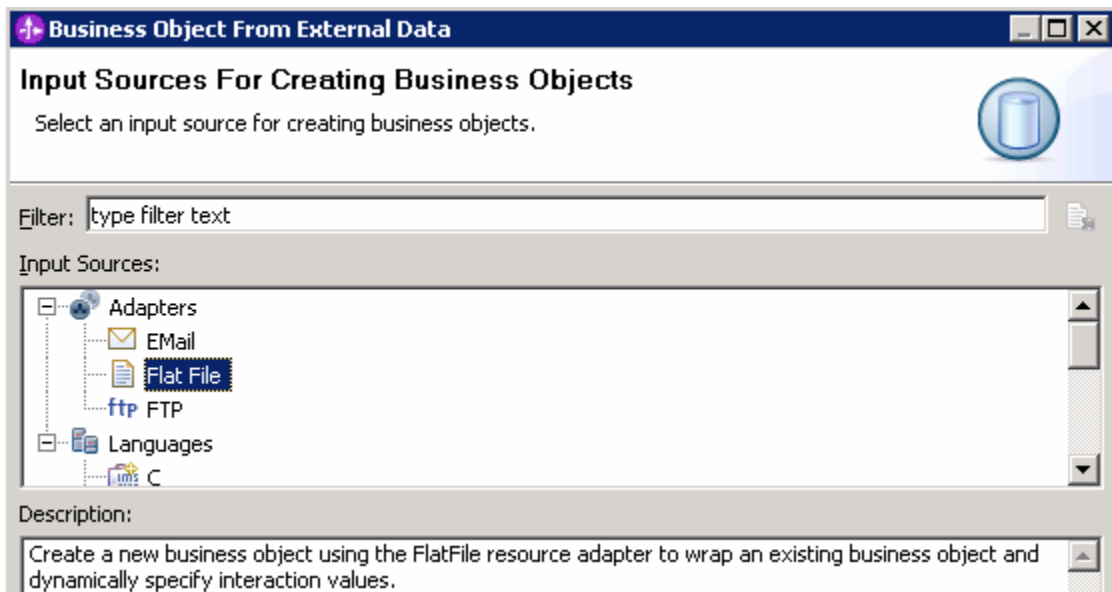


IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

A business object, called DFHCOMMAREA, is created in the module



- ___ 3. Generate a wrapper business object definition. Wrapper business object definitions wrap existing business object definitions with additional function.
 - ___ a. In the Business Integration section of the window, right-click the module and select **New > Business Object from External Data**
 - ___ b. From the External Data window, select **Adapters > Flat File** under Available Types:



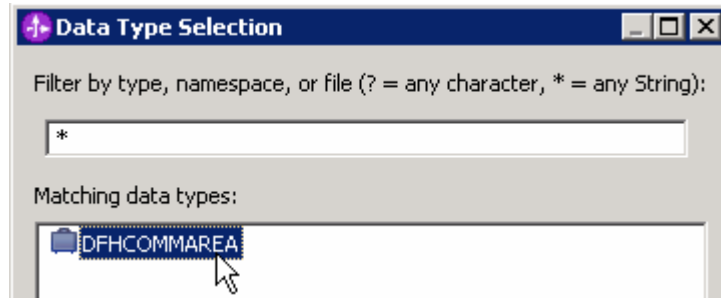
- ___ c. From the next screen, ensure that the correct module, **FFCOBOLInboundModule**, is selected and click **Next**

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

- ___ d. Provide the below values in Business Object Properties screen:
 - 1) Data type: click **Browse....** A Data Type selection window is opened

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

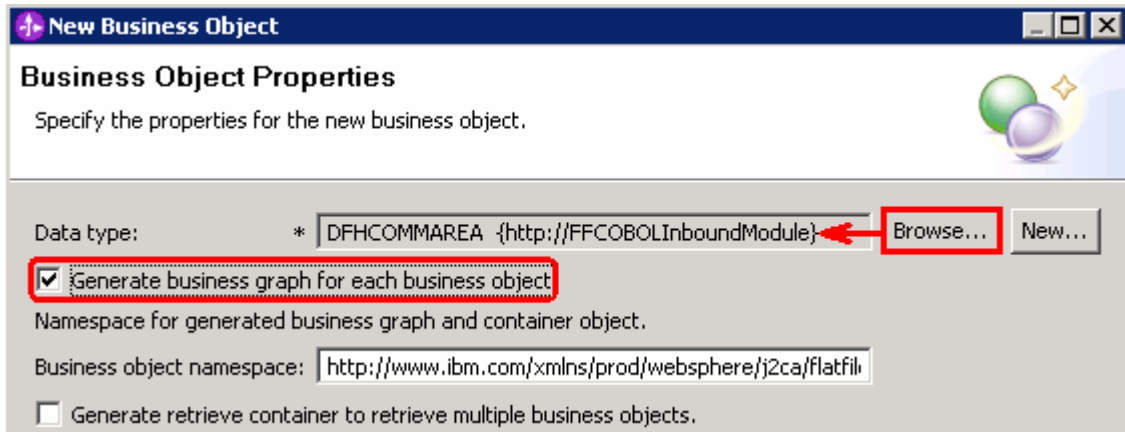
2) Select **DFHCOMMAREA** under Matching data types



3) Click **OK**

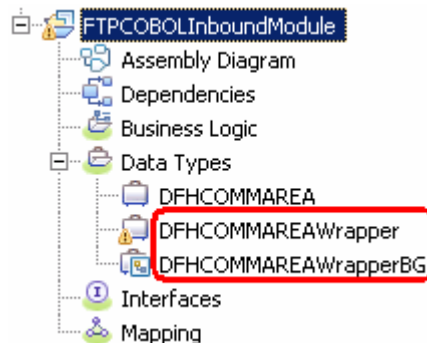
___ e. To generate a business graph, select the **Generate business graph for each business object** check box

___ f. **Do not** select the **Generate retrieve container to retrieve multiple business objects** check box. This new options is used and explained in outbound scenario



___ g. Click **Finish**

A wrapper business object and a business graph, called **DFHCOMMAREAWrapper** and **DFHCOMMAREAWrapperBG**, as shown in the figure below, are listed for the current module in the Business Integration window

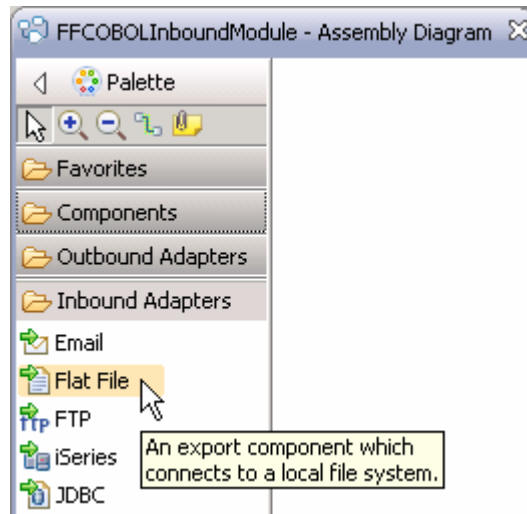


IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

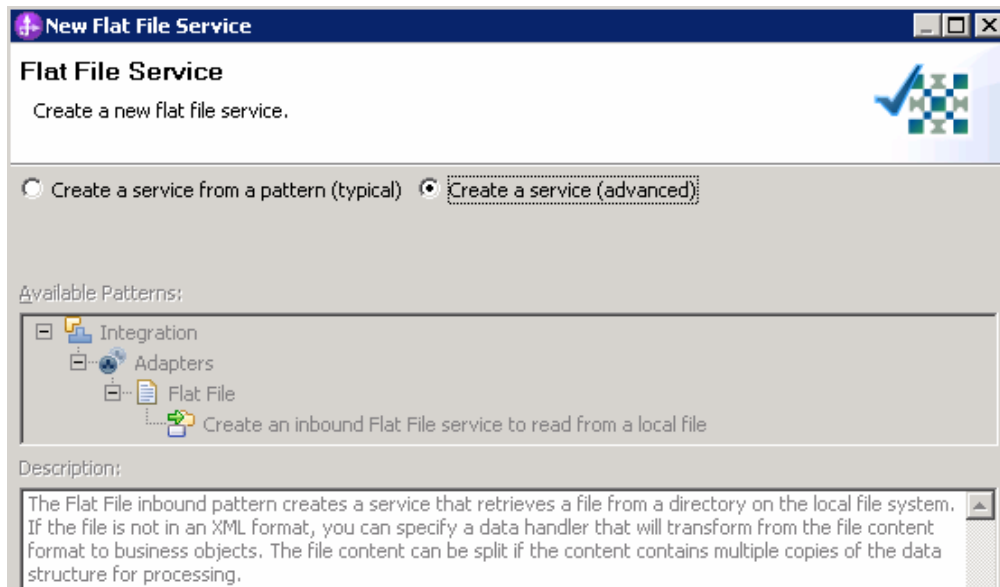
2.2. Configure inbound using the external service wizard

In this part of the lab you will use the new external service wizard and use the new 'COBOL, C, or PL/I data binding' to create the required artifacts to test the inbound scenario.

- ___ 1. Ensure that the FFCOBOLInboundModule assembly diagram is opened in Assembly editor
 - ___ a. In the Business Integration view, expand FFCOBOLInboundModule and double-click Assembly Diagram
- ___ 2. To start External Service from the Palette:
 - ___ a. From the **Palette** on the left side of Assembly Diagram, click **Inbound Adapters**:
- ___ 3. Under Inbound Adapters, click the **Flat File** and then click the empty canvas of the assembly diagram



- ___ 4. From the Flat File Service screen, select Create a service (advanced)



IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

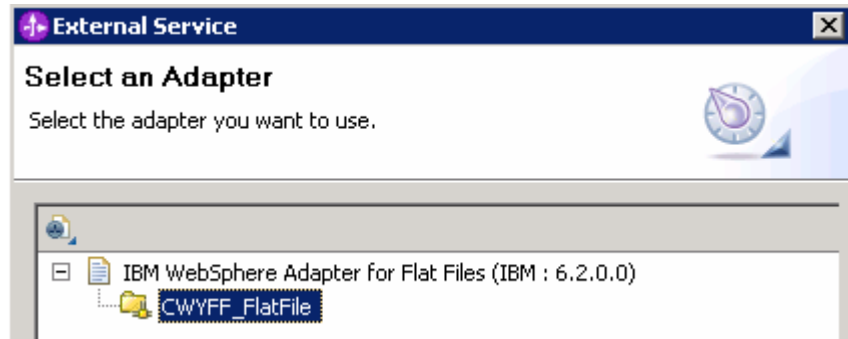
___ a. Click **Next**

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.

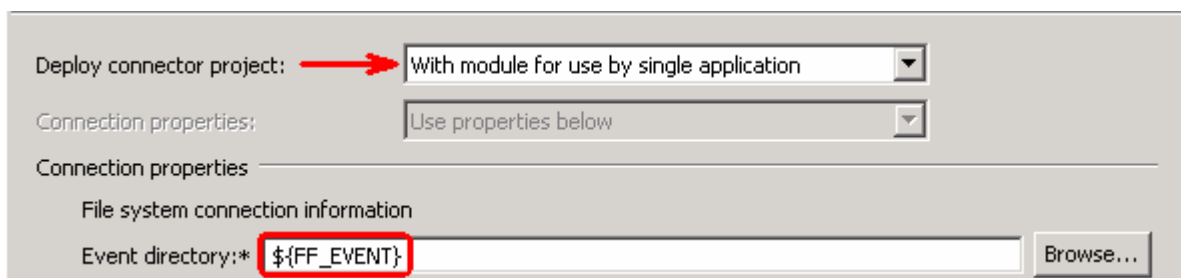
Select **Adapters > Flat File** and click **Next**

- ___ 5. On the Select an Adapter screen, select **IBM WebSphere Adapter for Flat Files (IBM : 6.2.0.0) > CWYFF_FlatFile** and click **Next**



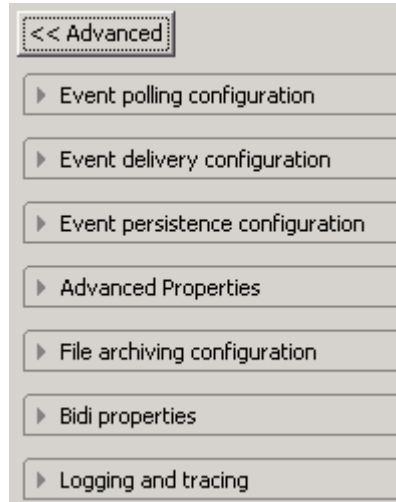
- ___ 6. Service Configuration Properties:
- ___ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected
- ___ b. Under Connection Configuration, for Event directory, enter **\${FF_EVENT}**:

Note: In FlatFile adapter, you have to give directory values (Event directory, Archive directory, Output directory and so on, needed values in activation spec and managed connection factory) while running external service wizard. Presently these values can be changed in the module and re-deploy it (or) can be changed in deployed application from administration console. This is like hard coding the values of these properties. All local directory (including event and archive) properties in ActivationSpec can be filled with WebSphere environment variables. This way hard coding of directory paths for these properties can be avoided. In WebSphere Process Server V6.1 and above, you can declare these values as environment variables 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 6.2 – LAB EXERCISE

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



You can click each of the configuration 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: In this part of the lab, you will not use any JNDI instead use adapter's in-memory representation of event table to store all the necessary information

Note: 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. Advanced properties:

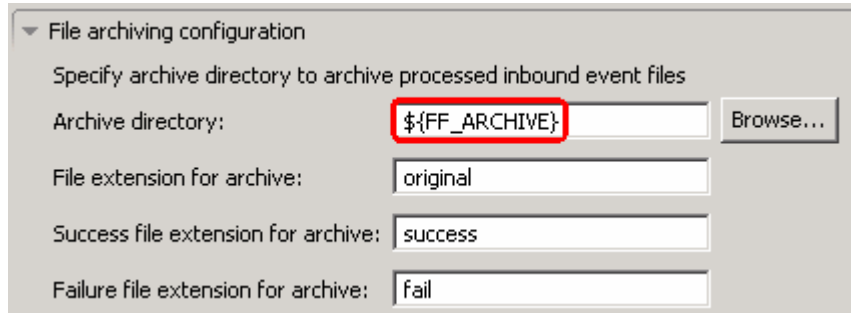
- 1) **Poll subdirectories in event directory:**

Note: '**Poll subdirectories in event directory**' is the new Boolean property added in V6.2. 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'. Refer to '**Log and confidential trace lab**' for more details on this new feature

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

__ h. File archiving configuration:

1) Archive directory: **`\${FF_ARCHIVE}`**

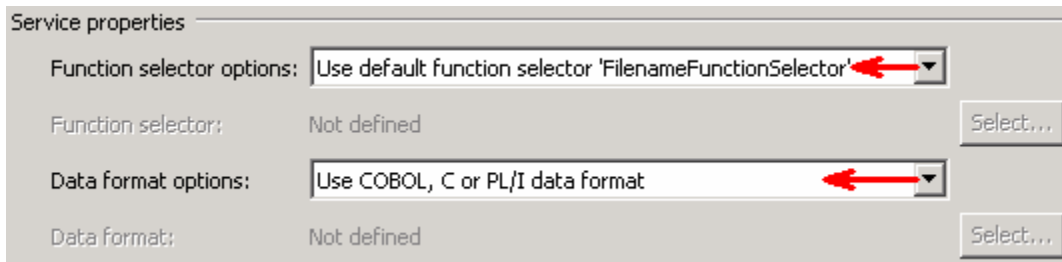


The screenshot shows a dialog box titled "File archiving configuration". It contains the following fields and controls:

- Archive directory: Browse...
- File extension for archive:
- Success file extension for archive:
- Failure file extension for archive:

___ 7. Under Service properties, for **Function selector options**, select **Use default function selector 'FilenameFunctionSelector'** from the drop down list

___ 8. For **Data format options**, select **Use COBOL, C or PL/I data format** from the drop down list



The screenshot shows a dialog box titled "Service properties". It contains the following fields and controls:

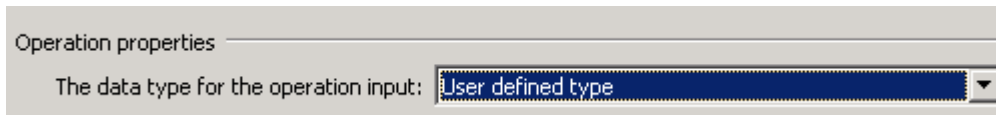
- Function selector options: (with a red arrow pointing to the dropdown arrow)
- Function selector: Not defined (with a Select... button)
- Data format options: (with a red arrow pointing to the dropdown arrow)
- Data format: Not defined (with a Select... button)

___ 9. 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 emitCOBOLFile operation:

___ 10. From the Operations screen, click **Add...**

___ a. Add Operation window is opened. Select **User Defined Type** for the Data type and click **Next**



The screenshot shows a dialog box titled "Operation properties". It contains the following field and control:

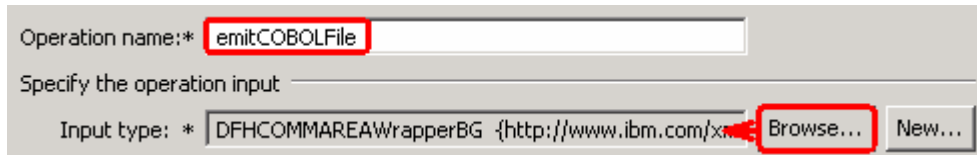
- The data type for the operation input: (with a dropdown arrow)

___ 11. For **Operation name**, enter any name, for Ex: **emitCOBOLFile**

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

- ___ 12. For **Input type**, click **Browse...** and select **DFHCOMMAREAWrapperBG** from the Data Type Selection window and click **OK**

Note: You can also define the business object from this screen by selecting **New**, but you can not be able to define the wrapper or business graph. So, for this driver, you should define the wrapper using External Data wizard as explained in Part 2.1 and select it here.

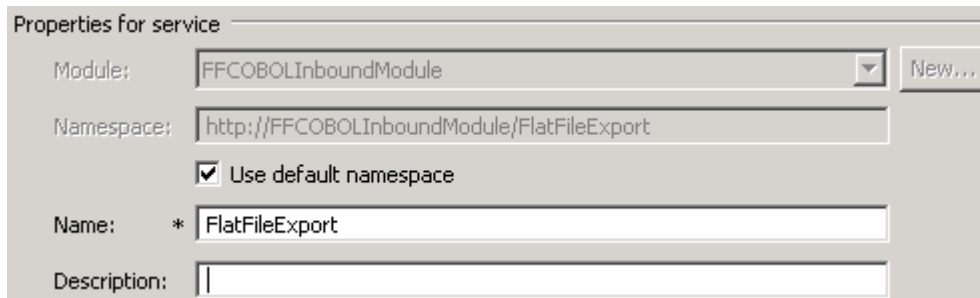


- ___ a. Click **Finish**. The above defined operation, **emitCOBOLFile**, is populated under Operations list



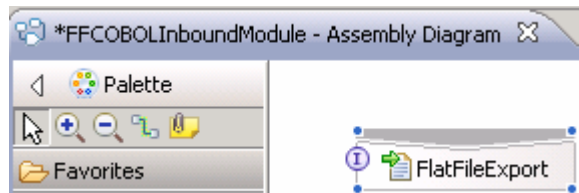
- ___ b. Click **Next** from Operations screen

- ___ 13. From Generate Service screen, accept the default value, **FlatFileExport**, for **Name**



- ___ a. Click **Finish**

- ___ 14. The Assembly diagram for FFCOBOLInboundModule is opened with an Export component, **FlatFileExport**:



- ___ a. Save (**Ctrl + S**) changes to your assembly diagram


IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

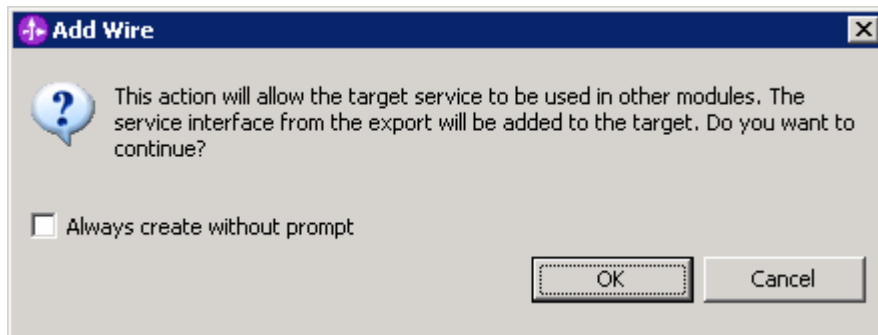
2.3. Add Java component

In this part of the lab, you will add a Java component and then wire the component to the existing Export interface. The Java component is your endpoint.

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



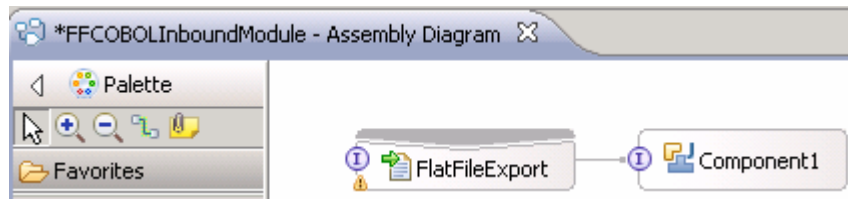
- ___ b. Click **Java** and then click the empty space of FFCOBOLInboundModule assembly diagram. This will place a new component, **Component1** on the assembly diagram.
- ___ 3. Wire the FlatFileExport to the Component1
 - ___ a. Select the **wire** () icon from the Palette
 - ___ b. Click **FlatFileExport**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



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

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Your assembly diagram for FFCOBOLInboundModule will look like this:



- ___ e. Right-click **Component1** and select **Generate Implementation** from the pop-up menu
- ___ f. On the **Generate Implementation** panel, select **default package**, and click **OK**
- ___ g. **Component1Impl.java** is opened in Assembly editor. Scroll down to the method **emitCOBOLFile** that needs to be implemented and add this code under that method:

Note: The Java code is also available at <FFFILES>\COBOLJavaCode.txt

```

System.out.println("*****ENDPOINT
emitCOBOLFile*****");
DataObject wrapper =
emitCOBOLFileInput.getDataObject("DFHCOMMAREAWrapper");
String filename = wrapper.getString("fileName");
System.out.println("File Name : "+filename);
DataObject customer = wrapper.getDataObject("Content");
String customernumber = customer.getString("CustomerNumber");
System.out.println("NAME-----> "+customernumber);
String firstname = customer.getString("FirstName");
System.out.println("First Name--> "+firstname);
String lastname = customer.getString("LastName");
System.out.println("Last Name--> "+lastname);
String street = customer.getString("Street");
System.out.println("Street--> "+street);
String city = customer.getString("City");
System.out.println("City-----> "+city);
String country = customer.getString("Country");
System.out.println("Country----> "+country);
String phone = customer.getString("Phone");
System.out.println("Phone-----> "+phone);
String postalcode = customer.getString("PostalCode");
System.out.println("Postal Code----> "+postalcode);

```

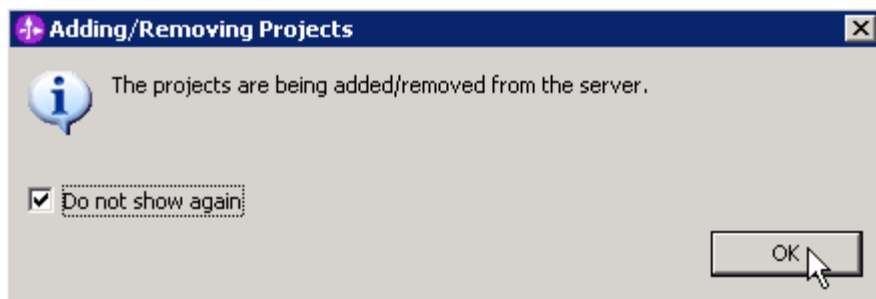
- ___ h. Save (**Ctrl + S**) and close Component1Impl.java
- ___ i. Save (**Ctrl + S**) and close Assembly diagram: FFCOBOLInboundModule

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

2.4. Test inbound scenario

In this part of the lab, you will use the WebSphere Process Server Test Environment to test the SCA application inbound processing for the COBOL copybook scenario.

- ___ 1. Add the project to the WebSphere Test Environment server
 - ___ a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects...** from the pop-up menu
 - ___ b. From the Add and Remove Projects window, select **FFCOBOLInboundModuleApp** under Available projects panel and click **Add >**
 - ___ c. You will now see the **FFCOBOLInboundModuleApp** 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
 - ___ e. Click **OK** from Adding/Removing Projects pop-up. Optionally, you can select 'Do not show again' so that you are not required to do this next time when you start the test client



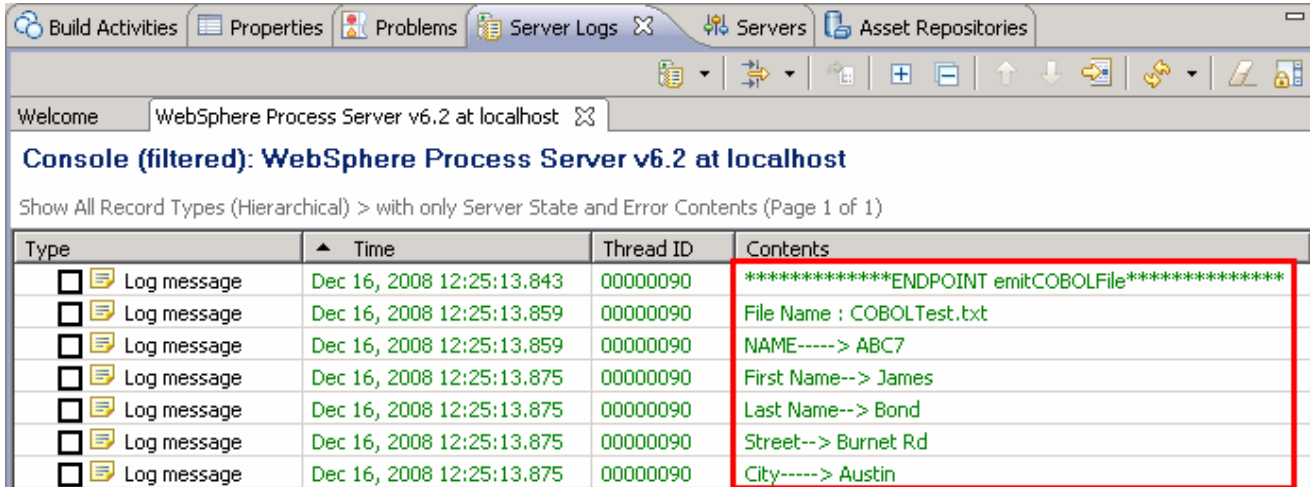
- ___ 2. Test the application by providing input files:

Note: For your convenience, **COBOLTest.txt** is placed in **<FFFILES>**.

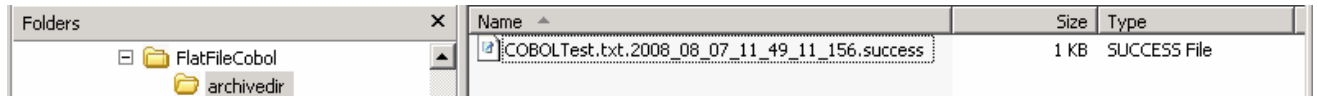
- ___ a. Copy the **COBOLTest.txt** file from **<FFFILES>** to **<EVENT_DIR>**. The adapter will poll the copied file from the event directory and will transfer it to the archive directory

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

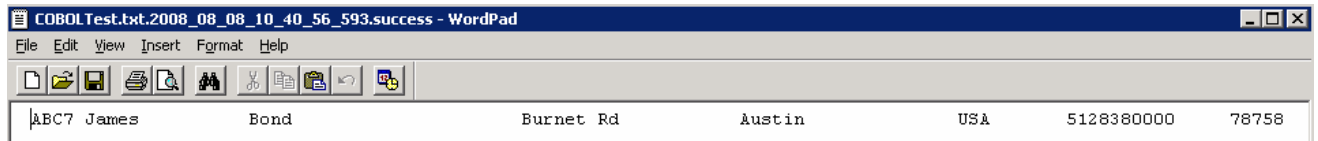
- __ b. It will pass through the **emitFlatFileBG** method and you should see this message in your **Server Logs** view (or SystemOut.log):



- __ 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



- __ d. Open the file using note pad or word pad and verify the contents of it



___ 3. Restore the Sever Configuration

- __ a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects...** from the pop-up menu
- __ b. Select **FFCOBOLInboundModuleApp** 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 6.2 – LAB EXERCISE

Part 3: Outbound scenario

In this part, you will use the external data wizard to generate a business object definition for a COBOL program source file. After you have generated the business object definition, you can optionally rerun the external data wizard to generate a wrapper business object definition from the generated business object.

You will make use of the new 'COBOL, C, or PL/I data binding' introduced in V6.2, while running the External service wizard to generate the required artifacts to convert business objects to COBOL copybook files during outbound processing.

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

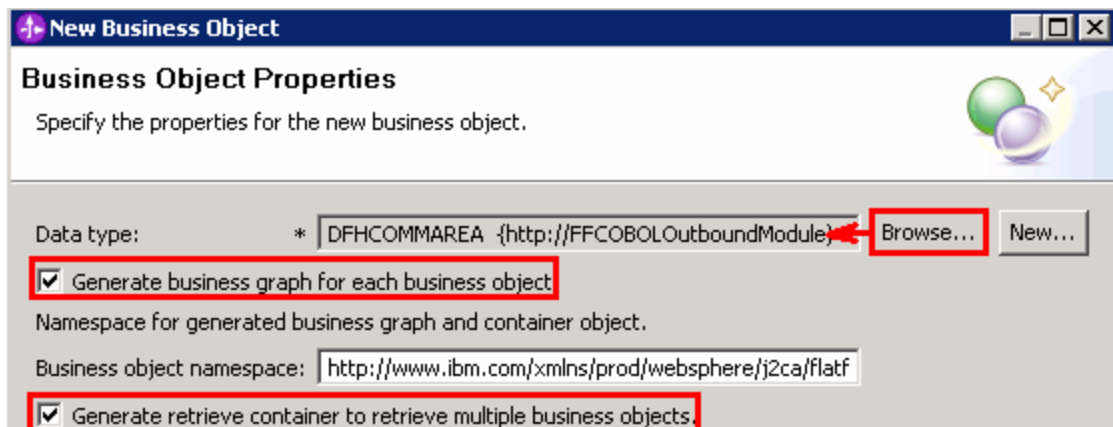
3.1. Prepare for the outbound scenario

In this part of the lab you will create a module, use the external data wizard to generate a business object definition for a COBOL program source file. After you have generated the business object definition, you can optionally rerun the external data wizard to generate a wrapper business object definition from the generated business object.

- ___ 1. Create FFCOBOLOutboundModule
 - ___ a. From the Business Integration window, right-click and select **New > Module**
 - ___ b. From the New Module window, enter **FFCOBOLOutboundModule** for the Module Name
 - ___ c. Ensure that the box next to **Open module assembly diagram** is checked and then click **Finish**

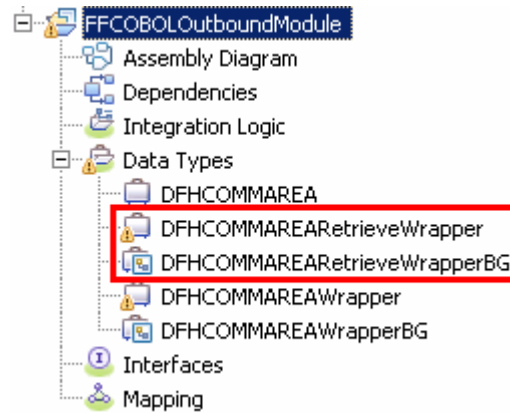
You will now see a new module, FFCOBOLOutboundModule, created from your Business Integration window

- ___ 2. Follow the instructions in **Steps 2 and 3 of Part 2.1** to generate business object and a wrapper definition under **FFCOBOLOutboundModule**
- ___ 3. Generate a wrapper business object definition for retrieve operation.
 - ___ a. In the Business Integration section of the window, right-click the module and select **New > Business Objects From External Data**
 - ___ b. From the External Data window, select **Adapters > Flat File** under Available Types:
 - ___ c. From the next screen, ensure that the correct module, **FFCOBOLOutboundModule**, is selected and click **Next**
 - ___ d. Provide the below values in Business Object Properties screen:
 - 1) Data type: click **Browse....** A Data Type selection window is opened
 - 2) Select **DFHCOMMAREA** under Matching data types and click **OK**
 - ___ e. To generate a business graph, select the **Generate business graph for each business object** check box
 - ___ f. Select the **Generate retrieve container to retrieve multiple business objects** check box



IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

- ___ g. Click **Finish**
- ___ h. A wrapper business object and a business graph, called **DFHCOMMAREARetrieveWrapper** and **DFHCOMMAREARetrieveWrapperBG**, as shown in the figure below, are listed for the current module in the Business Integration window. This **DFHCOMMAREARetrieveWrapperBG** is used during External Service wizard in the next part for retrieve operation.

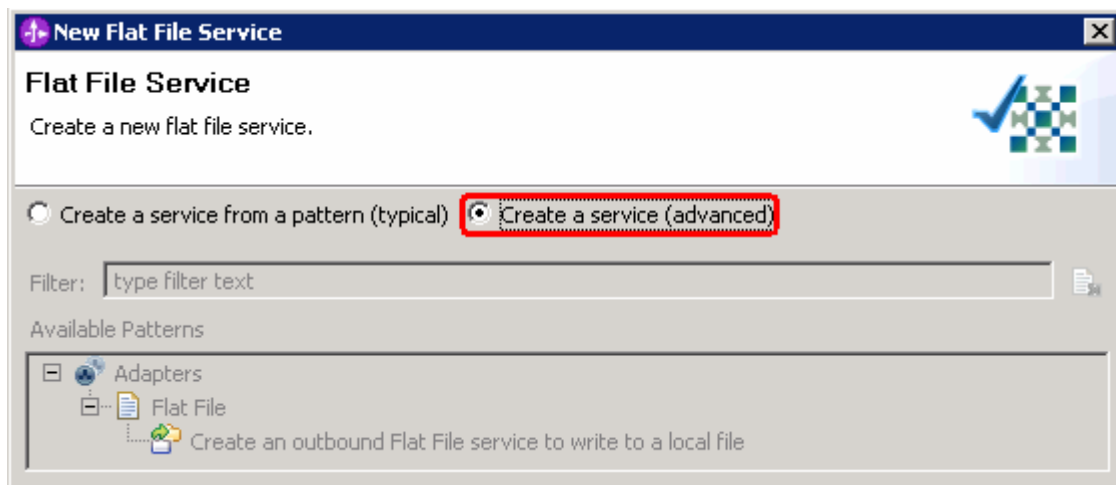


IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

3.2. Configure outbound using external service wizard

In this part of the lab you will use the new external service wizard and use the new 'COBOL, C, or PL/I data binding' to create the required artifacts to test the outbound scenario.

- ___ 1. Ensure that the FFCOBOLOutboundModule assembly diagram is opened in Assembly editor
 - ___ a. In the Business Integration view, expand FFCOBOLOutboundModule and double-click Assembly Diagram
- ___ 2. To start External Service from the Palette:
 - ___ a. From the **Palette** on the left side of Assembly Diagram, click **Outbound Adapters**
 - ___ b. Under Outbound Adapters, click the **Flat File** and then click the empty canvas of the assembly diagram. The New Flat File Service wizard is opened
- ___ 3. From the Flat File Service screen, select **Create a service (advanced)**



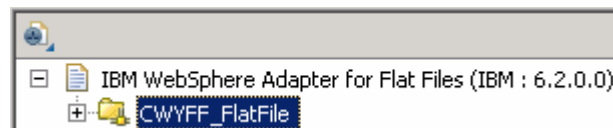
- ___ a. Click **Next**

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**.

- ___ 4. On the Select an Adapter screen, expand **IBM WebSphere Adapter for Flat Files (IBM : 6.2.0.0)** and select **CWYFF_FlatFile**

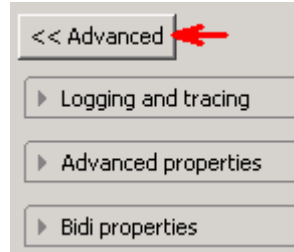


- ___ a. Click **Next**

- ___ 5. Service Configuration Properties:

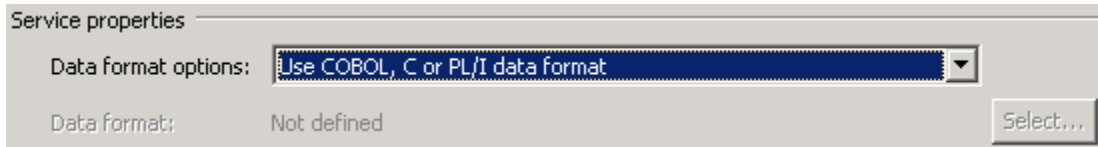
IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

- ___ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected
- ___ b. For **Output Directory** enter **\${FF_OUT}**
- ___ c. Click **Advanced >>** to see the hidden advanced properties that can be configured:



You can click each of the configuration and review the options available under it. This lab guides you through some of the important and required configurations.

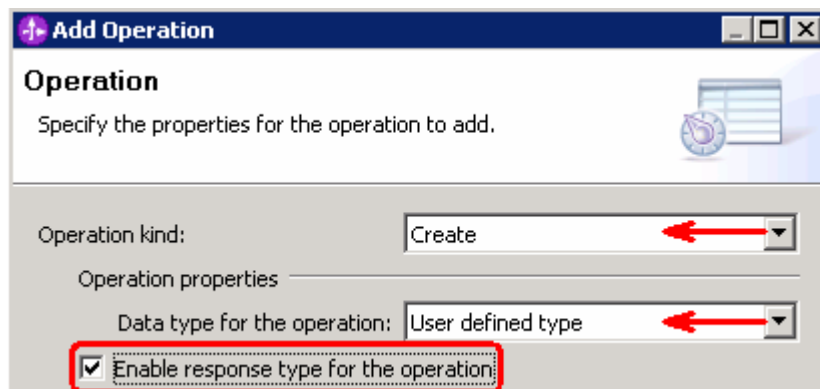
- ___ 6. For **Data format options**, select **Use COBOL, C or PL/I data format** from the drop down list



- ___ 7. 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 createCOBOL operation:

- ___ 8. From the Operations screen, click **Add...**
 - ___ a. In the Add Operation window, select **Create** from the drop down menu for Operation kind
 - ___ b. Select **User Defined Type** for the Data type
 - ___ c. Select the check box next to **Enable response type for the operation**



- ___ d. Click **Next**

- ___ 9. For **Operation name**, enter any name, for Ex: **createCOBOL**

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

- ___ 10. For Input type, click **Browse** and select **DFHCOMMAREAWrapperBG** from the Data Type Selection window

The screenshot shows a form for defining an operation. The 'Operation name' field contains 'createCOBOL'. Under 'Specify the operation input', the 'Input type' is set to 'DFHCOMMAREAWrapperBG' and the 'Browse...' button is highlighted with a red box. Under 'Specify the operation output', the 'Output type' is set to 'CreateResponse'.

- ___ a. Click **Finish**. The above defined operation, **createCOBOL**, is populated under Operations list

The screenshot shows a list of operations. The entry 'createCOBOL' is visible, followed by its full path: '{http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperb'. An 'Add...' button is located to the right of the list.

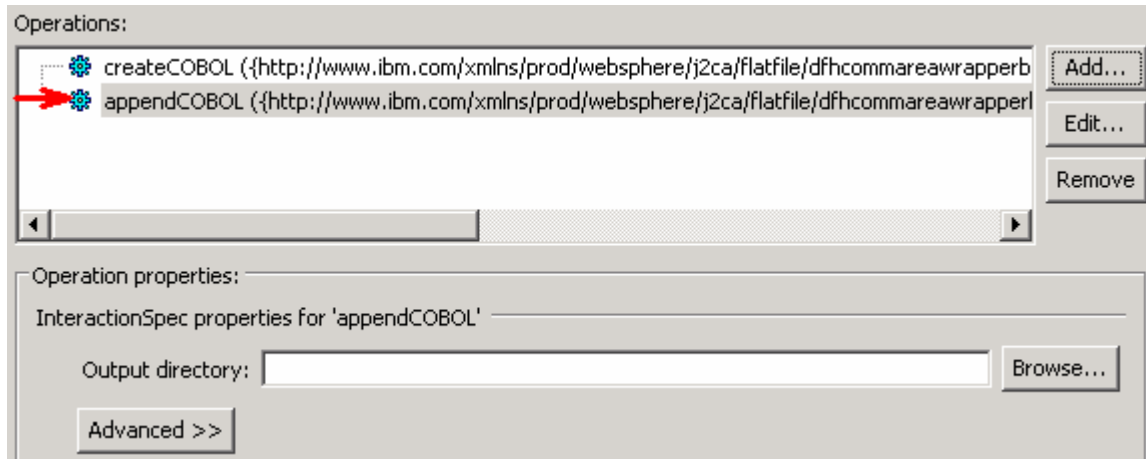
Define appendCOBOL operation:

- ___ 11. From the Operations screen, click **Add...**
- ___ a. In the Add Operation window, select **Append** from the drop down menu for Operation kind
 - ___ b. Select **User Defined Type** for the Data type
 - ___ c. Select the check box next to **Enable response type for the operation**
 - ___ d. Click **Next**
 - ___ e. For **Operation name**, enter **appendCOBOL**
 - ___ f. For Input type, click **Browse** and select **DFHCOMMAREAWrapperBG** from the Data Type Selection window

The screenshot shows a form for defining an operation. The 'Operation name' field contains 'appendCOBOL'. Under 'Specify the operation input', the 'Input type' is set to 'DFHCOMMAREAWrapperBG' and the 'Browse...' button is highlighted with a red box. Under 'Specify the operation output', the 'Output type' is set to 'AppendResponse'.

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

__ g. Click **Finish**. The above defined operation, **appendCOBOL**, is populated under Operations list



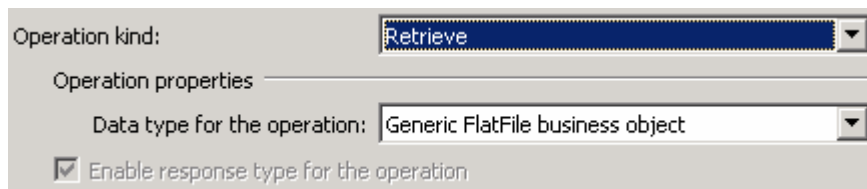
Define retrieveCOBOL operation:

___ 12. Click **Add...** to open Add Operation window

__ a. For **Operation kind**, select **Retrieve** from the drop down list

__ b. For **Data type for operation**, select **Generic FlatFile business object** from the drop down list

__ c. Note that the box next to **Enable response type for the operation** is checked by default



__ d. Click **Next**

In the Add Operation window, under Specify the operation input, you will see the Input type **FlatFile** (because you have selected **not to** have business graph (BG)) and you will also see the Output type **RetrieveResponseWrapper** under 'Specify the operation output' (because the Output was selected by default).

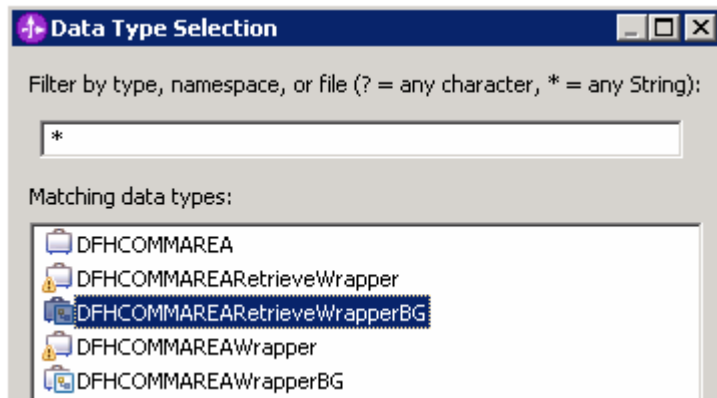
__ e. For Operation name, enter **retrieveCOBOL**

__ f. Define Output type:

- 1) Under **Specify the operation output**, click **Browse...** next to **Output type** to open a Data Type Selection window

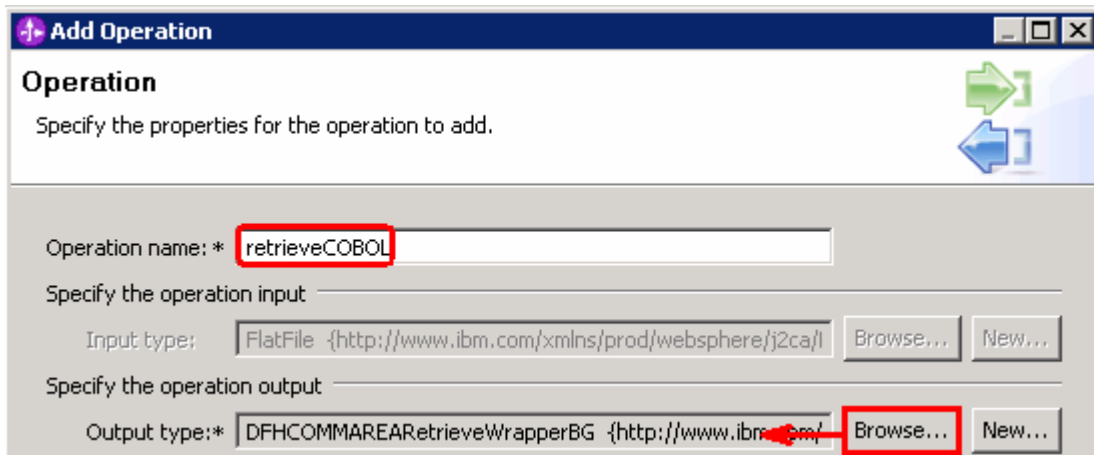
IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

- 2) From the Data Type Selection window, select **DFHCOMMAREARetrieveWrapperBG** under Matching data types



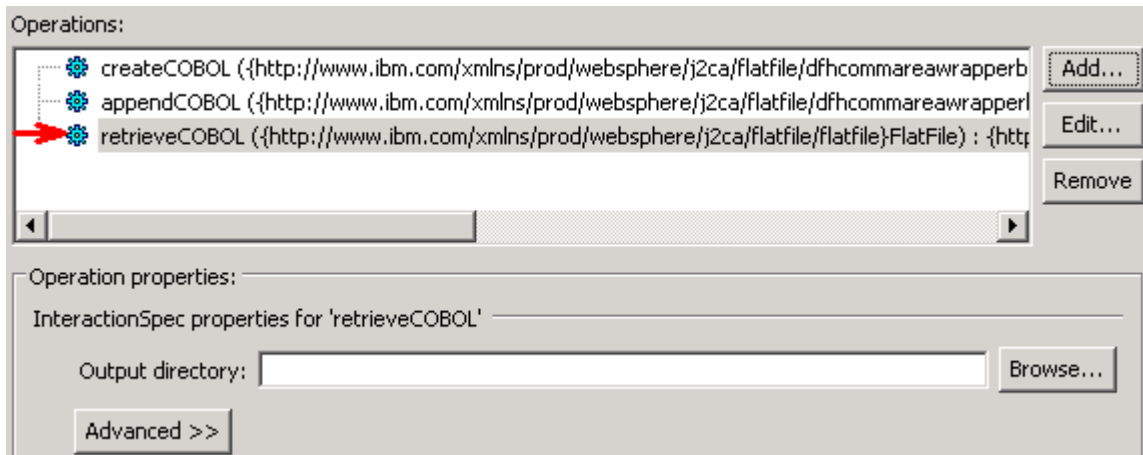
- 3) Click **OK**

__ g. You should now see this:



- __ h. Click **Finish** from the Add Operation window

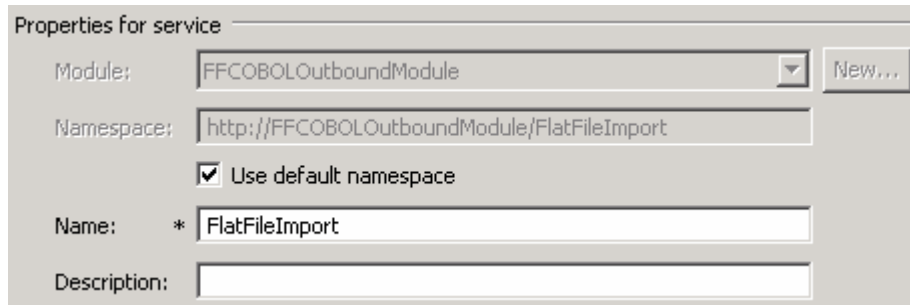
- __ i. You can click **Advanced >>** under 'InteractionSpec properties for retrieveCOBOL' to review the properties available at Interaction spec level



IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

___ j. Click **Next** from Operations screen

___ 13. From Generate Service screen, accept the default value, **FlatFileImport**, for **Name**



Properties for service

Module: FFCOBOLOutboundModule New...

Namespace: http://FFCOBOLOutboundModule/FlatFileImport

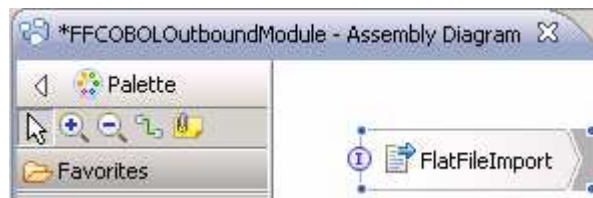
Use default namespace

Name: * FlatFileImport

Description:

___ a. Click **Finish**

___ 14. The Assembly diagram for FFCOBOLOutboundModule is opened with an Export component, **FlatFileImport**



___ 15. Save (**Ctrl + S**) changes to your assembly diagram

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

3.3. Test outbound scenario

In this part of the lab, you will use the WebSphere Process Server test environment to test the SCA application outbound processing for the COBOL copybook create scenario.

- ___ 1. Start WebSphere Process Server (if not started already)
 - ___ a. From the **Servers** view of WebSphere Integration Developer, right click **WebSphere Process Server v6.2** and select Start from the pop-up menu
 - ___ b. Wait until the server status shows as **Started**
- ___ 2. Add the project to the WebSphere test environment server
 - ___ a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects...** from the pop-up menu
 - ___ b. From the Add and Remove Projects window, select **FFCOBOLOutboundModuleApp** under Available projects panel and click **Add >**
 - ___ c. You will now see the **FFCOBOLOutboundModuleApp** added to the **Configured projects**
 - ___ d. Click **Finish** and wait until the project is being published onto the server. The server is started in Debug mode if it is not already started before
- ___ 3. Open the test client for the module
 - ___ a. From the Business Integration perspective, right-click the **FFCOBOLOutboundModule** and select **Test > Test Component**
 - ___ b. The **FFCOBOLOutboundModule_Test** window is opened in the Assembly editor

You have three operations that were defined in the previous part in this module:

- createCOBOL
- appendCOBOL
- retrieveCOBOL

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

Test Create operation:

- ___ 4. Under **Detailed Properties**, note that the Operation is **createCOBOL**
 - ___ a. Fill out the fields for Initial request parameters:
 - ___ b. For **fileName**, enter any name, for Ex: **COBOLTest.txt**
 - ___ c. For **includeEndBODelimiter**, enter **###**

► **General Properties**

▼ **Detailed Properties**

Configuration: Default Module Test

Module: FFCOBOLOutboundModule

Component: FlatFileImport

Interface: FlatFileImport

Operation: createCOBOL ←

Initial request parameters

Name	Type	Value
createCOBOLInput	DFHCOMMAREAWrapperBG	✓
verb	verb<string>	✓ CREATE
DFHCOMMAREAWrapper	DFHCOMMAREAWrapper	✓
directoryPath	string	✓
fileName	string	✓ COBOLTest.txt
chunkFileName	string	✓
fileContentEncoding	string	✓
includeEndBODelimiter	string	✓ ###
stagingDirectory	string	✓

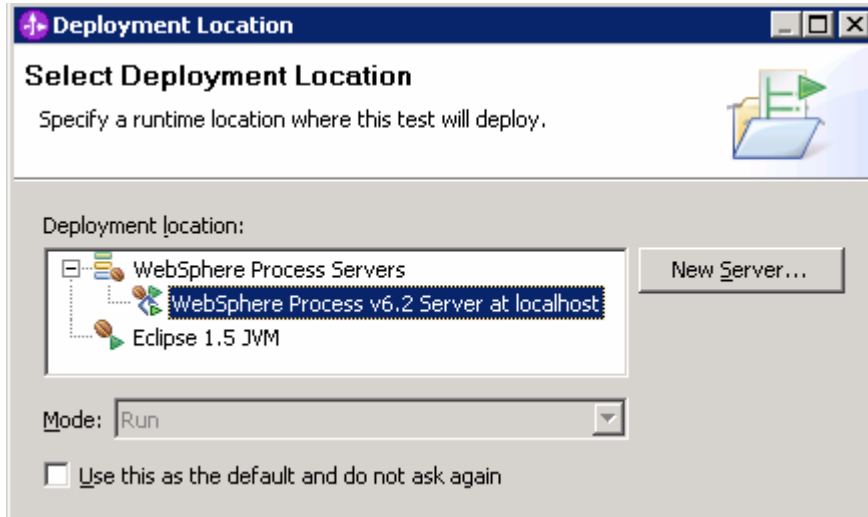
- ___ d. Enter the values of your choice for the fields under **Content**

Content	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber <string>	✓ ABC7
FirstName	FirstName <string>	✓ James
LastName	LastName <string>	✓ Bond
Street	Street <string>	✓ Burnet Rd
City	City <string>	✓ Austin
Country	Country <string>	✓ USA
Phone	Phone <string>	✓ 5128380000
PostalCode	PostalCode <string>	✓ 78758

- ___ e. Click **Continue** button under Events

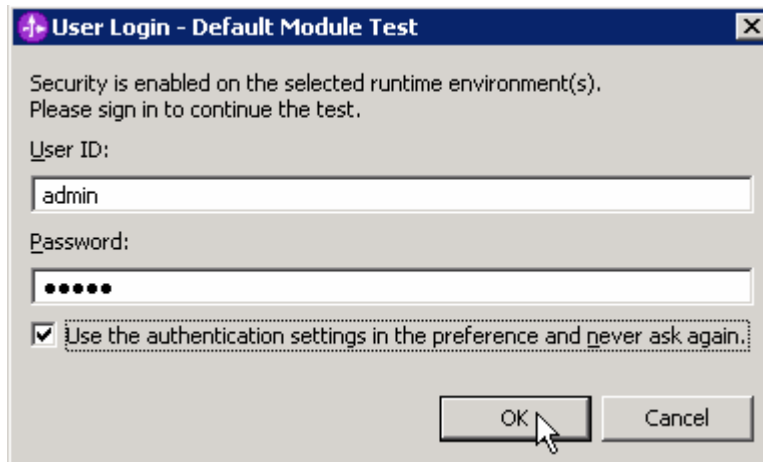
IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

- ___ f. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server v6.2 at localhost** and click **Finish**



- ___ g. Provide Administrator User ID and Password

- 1) Optionally, select the box 'Use the authentication settings in the preference and never ask again' so that you do not have to enter the credentials next time when you start the test client



- ___ h. Click **OK** from Adding/Removing Projects pop-up, if asked. Optionally, you can select 'Do not show again' so that you are not required to do this next time when you start the test client

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

___ i. Verify the results:

- 1) You should see a window similar to this, which contains the data you just entered in the previous steps:

The screenshot shows the IBM WebSphere Adapter 6.2 interface. On the left, the 'Events' pane shows a sequence of events: 'Invoke (FlatFileImport:createCOBOL)', 'Invoke started', 'Invoke (FlatFileImport:createCOBOL)', 'Return (FlatFileImport:createCOBOL)' (highlighted with a red box), and 'Invoke returned'. On the right, the 'Detailed Properties' pane shows the following information:

- Module: [FFCOBOLOutboundModule](#)
- Component: [FlatFileImport](#)
- Interface: [FlatFileImport](#)
- Operation: [createCOBOL](#)

Below the properties, the 'Return parameters:' section contains a table:

Name	Type	Value
createCOBOLOutput	CreateResponse	✓
filename	string	✓ COBOLTest.txt

- 2) To verify your test results, check the <OUT_DIR> directory which should contain a file with the name **COBOLTest.txt**

The screenshot shows a file explorer window with the following details:

- Address: C:\Labfiles62\FlatFileCobol\outdir
- Folders: FlatFileCobol
- Name: COBOLTest.txt (highlighted)

- 3) Double-click to open the file, **COBOLTest.txt** and it should contain the text entered under Content field in your test client

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

Test Append operation:

- ___ 5. Under **Detailed Properties**, note that the Operation is **appendCOBOL**
 - ___ a. Fill out the fields for Initial request parameters:
 - ___ b. For **fileName**, enter any name, for Ex: **COBOLTest.txt**
 - ___ c. For **includeEndBODelimiter**, enter **###**

► **General Properties**


▼ **Detailed Properties**

Configuration: Default Module Test

Module: FFCOBOLOutboundModule

Component: FlatFileImport

Interface: FlatFileImport

Operation: appendCOBOL 

Initial request parameters

Name	Type	Value
appendCOBOLInput	DFHCOMMAREAWrapperBG	✓
verb	verb<string>	✓ CREATE
DFHCOMMAREAWrapper	DFHCOMMAREAWrapper	✓
directoryPath	string	✓
fileName	string	✓ COBOLTest.txt
chunkFileName	string	✓
fileContentEncoding	string	✓
includeEndBODelimiter	string	✓ ###

- ___ d. Enter the values of your choice for the fields under **Content**

Content	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber<string>	✓ XYZ9
FirstName	FirstName<string>	✓ James
LastName	LastName<string>	✓ Doe
Street	Street<string>	✓ Duval Rd
City	City<string>	✓ Austin
Country	Country<string>	✓ USA
Phone	Phone<string>	✓ 5128380000
PostalCode	PostalCode<string>	✓ 78727

- ___ e. Click **Continue** button under Events

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

___ f. Verify the results:

- 1) You should see a window similar to this, which contains the data you just entered in the previous steps:

The screenshot shows the Event Viewer interface. The left pane displays a tree of events, with the following structure:

- Invoke (FlatFileImport:createCOBOL)
 - Invoke started
 - Invoke (FlatFileImport:createCOBOL)
 - Return (FlatFileImport:createCOBOL)
 - Invoke returned
- Invoke (FlatFileImport:appendCOBOL)
 - Invoke started
 - Invoke (FlatFileImport:appendCOBOL)
 - Return (FlatFileImport:appendCOBOL) - **Selected**
 - Invoke returned

The right pane shows the 'Detailed Properties' for the selected event:

- Module: [FFCOBOLOutboundModule](#)
- Component: [FlatFileImport](#)
- Interface: [FlatFileImport](#)
- Operation: [appendCOBOL](#)
- Return parameters:

Name	Type	Value
appendCOBOLou...	AppendResponse	✓
filename	string	✓ COBOLTest.txt

- 2) To verify your test results, open the COBOLTest.txt file from <OUT_DIR> directory and it should contain the text entered under Content field in your test client. The highlighted text is the appended content and note the end BO delimiter, ###, at the end of each COBOL BO.

The screenshot shows a Notepad window titled 'COBOLTest.txt' with the following content:

```

ABC7 James          Bond          Burnet Rd        Austin          USA
5128380000        78758        ###XYZ9 James    Doe             Duval Rd
Austin            USA          5128380000      78727        ###
    
```

The second and third lines of the table are highlighted in blue in the original image.

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

Test Retrieve operation:

- ___ 6. Under **Detailed Properties**, note that the Operation is **retrieveCOBOL**
 - ___ a. Fill out the fields for Initial request parameters:
 - ___ b. For **filename**, enter **COBOLTest.txt**
 - ___ c. **splitFunctionClassName**: **com.ibm.j2ca.utils.filesplit.SplitByDelimiter**
 - ___ d. **splitCriteria**: **###**

Initial request parameters

Name	Type	Value
retrieveFlatFileInput	FlatFile	✓
directoryPath	string	✓
fileName	string	✓ COBOLTest.txt
chunkFileName	string	✓
fileContentEncoding	string	✓
includeEndBODelimiter	string	✓
stagingDirectory	string	✓
chunkNumber	string	✓
generateUniqueFile	boolean	✓ false
createFileIfNotExists	boolean	✓ false
splitFunctionClassName	string	✓ com.ibm.j2ca.utils.filesplit.SplitByDelimiter
splitCriteria	string	✓ ###
deleteOnRetrieve	boolean	✓ false
archiveDirectoryForDeleteOnRetrieve	string	✓
Content	UnstructuredContent	✓
ContentType	string	✓
ObjectName	string	✓
AsText	string	✓
AsBinary	hexBinary	✓ 00

- ___ e. Click **Continue** button under Events

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

___ f. Verify the results:

- 1) You should see a window similar to this, which contains the data you just entered in the previous steps:

Return parameters

Name	Type	Value
retrieveCOBOLOutput	DFHCOMMAREARetrieveWrapperBG	✓
verb	verb <string>	✓
DFHCOMMAREARetrieveWrapper	DFHCOMMAREARetrieveWrapper	✓
Content	DFHCOMMAREA[]	68
Content[0]	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber <string>	✓ ABC7
FirstName	FirstName <string>	✓ James
LastName	LastName <string>	✓ Bond
Street	Street <string>	✓ Burnet Rd
City	City <string>	✓ Austin
Country	Country <string>	✓ USA
Phone	Phone <string>	✓ 5128380000
PostalCode	PostalCode <string>	✓ 78758
Content[1]	DFHCOMMAREA	✓
CustomerNumber	CustomerNumber <string>	✓ XYZ9
FirstName	FirstName <string>	✓ James
LastName	LastName <string>	✓ Doe
Street	Street <string>	✓ Duval Rd
City	City <string>	✓ Austin
Country	Country <string>	✓ USA
Phone	Phone <string>	✓ 5128380000
PostalCode	PostalCode <string>	✓ 78727

___ 7. Restore the Sever Configuration

- ___ a. Close the **FFCOBOLOutboundModule_Test** window and click **No** for the Save Resources window
- ___ b. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects...** from the pop-up menu
- ___ c. Select **FFCOBOLOutboundModuleApp** under Configured projects and click **< Remove**
- ___ d. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished

IBM WEBSPHERE ADAPTER 6.2 – 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 Data wizard to generate business object and wrapper definitions from a COBOL program source file. Then you continued to with External Service wizard available in WebSphere Integration Developer to generate other artifacts for inbound and then for outbound.

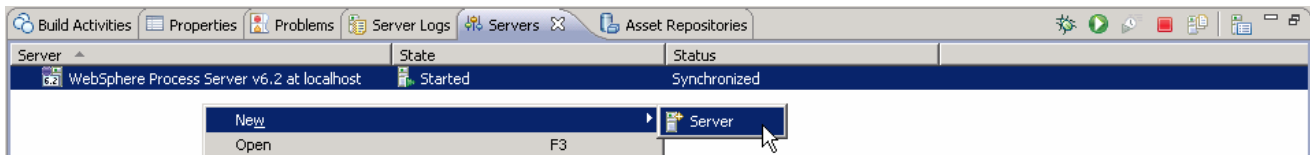
At the end of each part, you deployed and tested the adapter application for operations defined during the external service wizard.

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

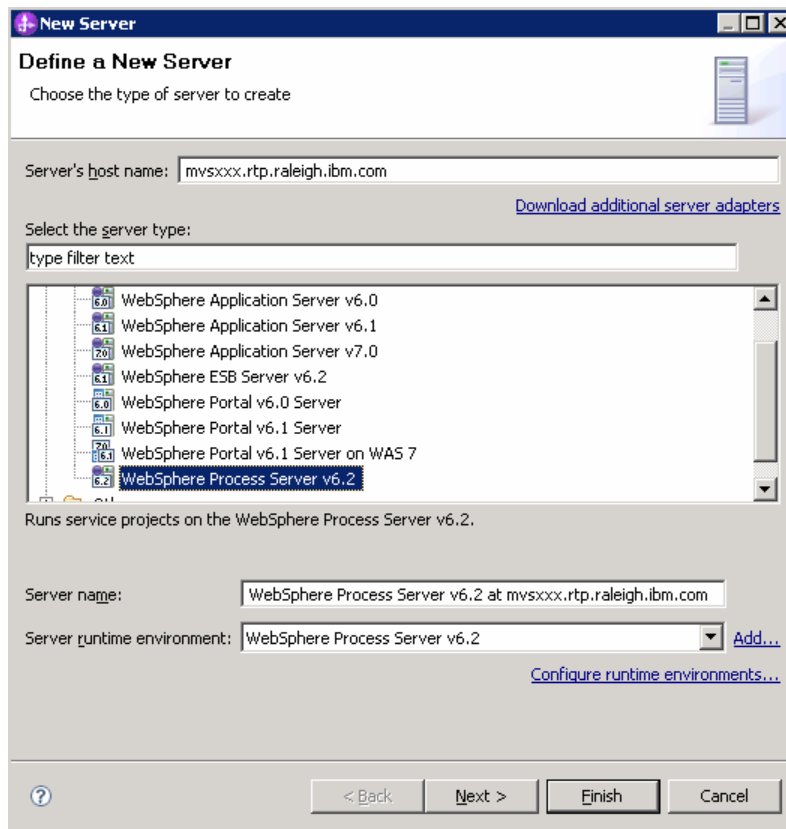
Task: Adding remote server to WebSphere Integration Developer test environment

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**.



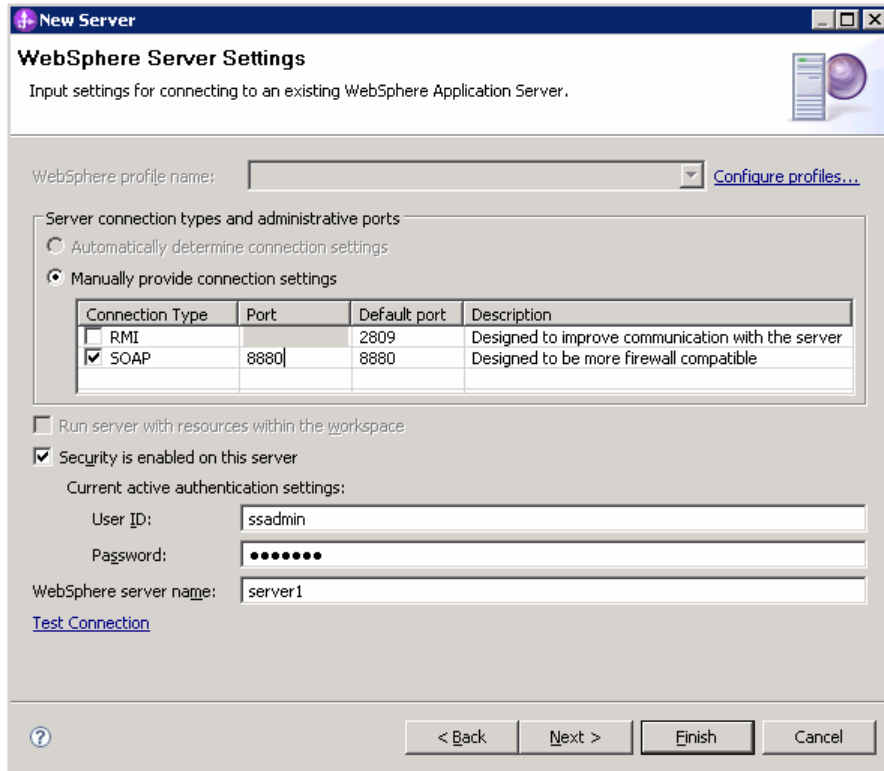
- ___ c. In the New Server dialog, specify the remote server's host name, **<HOSTNAME>**.
- ___ d. Ensure that the appropriate server type, **'WebSphere Process Server v6.2'** or **'WebSphere ESB Server v6.2'**, is highlighted in the server type list



- ___ e. Click **Next**.

IBM WEBSHERE ADAPTER 6.2 – 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 WEBSHERE ADAPTER 6.2 – 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
```

IBM WEBSHERE ADAPTER 6.2 – LAB EXERCISE

This page is left intentionally blank.