IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Flat File adapter outbound lab

What this exercise is about

The objective of this lab is to provide you with an understanding of the IBM WebSphere Adapter for Flat Files and outbound processing. In this lab you will deploy the WebSphere Adapter for Flat Files, using WebSphere Integration Developer, and integrate it with an SCA application that processes outbound requests to the file system.

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 with latest fixes

- Extract Labfiles62.zip to your C:\ (your root) drive

What you should be able to do

- Import Flat Files adapter RAR file into WebSphere Integration Developer

- Use External Service wizard to configure Activation Spec Properties, Resource Adapter Properties to generate Business Objects and other artifacts and then define your Data Binding and Data Handler, and Operations

- Deploy the adapter application onto the WebSphere Process Server test environment

- Test the deployed application using WebSphere Process Server test environment for both pass-through and non pass-through using different scenarios and patterns

- Restore the server configuration

Introduction
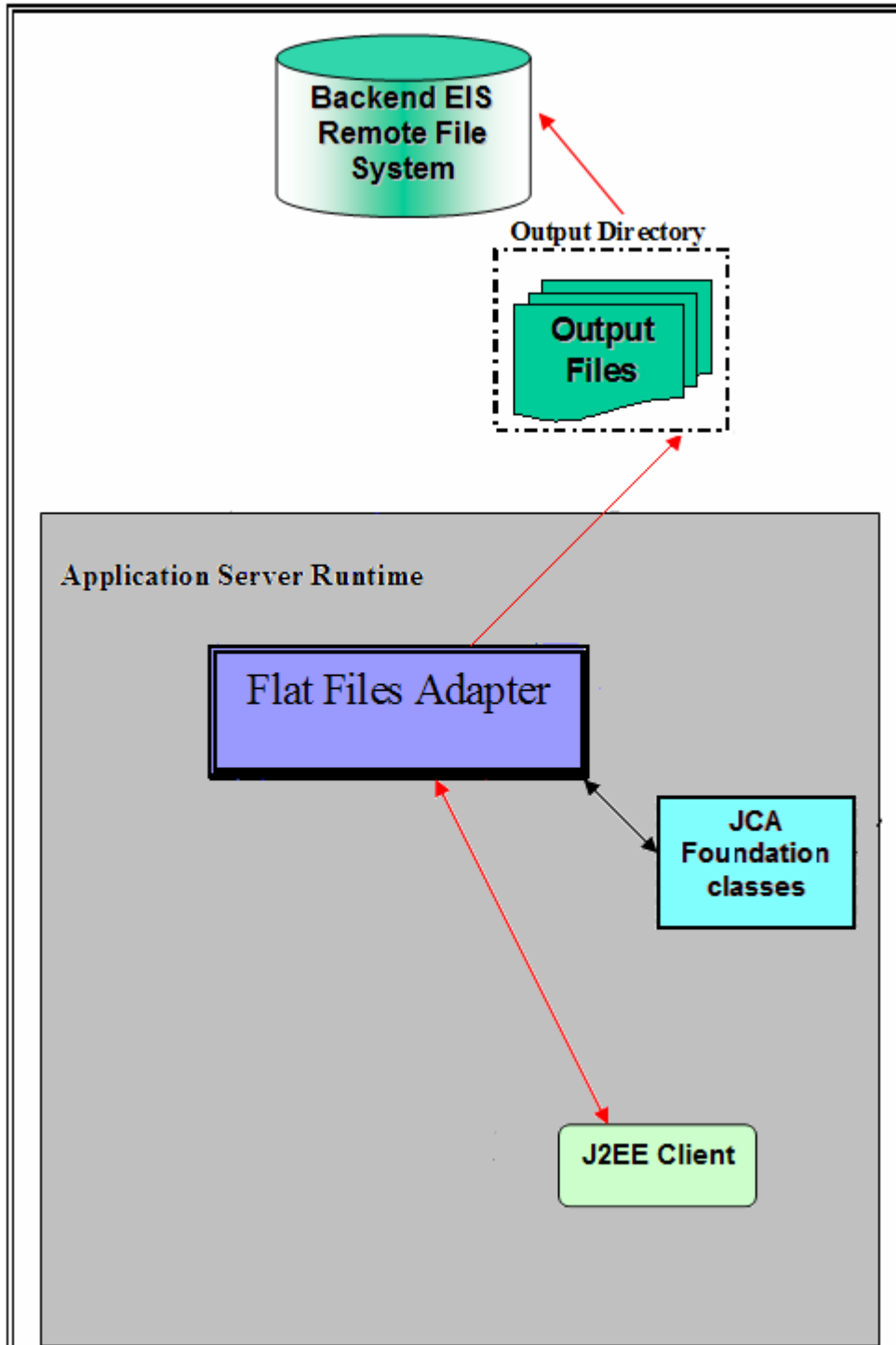
The request to the FF RA (Flat File Resource Adapter) from an SCA client is sent with the information required for creating the output file - the directory to create the output file, the name of the output file and the operation. The file name is treated as the key for the Flat Files request/response BO entity.

FF RA uses the application server's connection pool to establish connections with an EIS. The adapter creates a new file connection for every outbound operation and closes it after the operation is completed based on the EIS connectivity requirement. The requests initiated from any SCA client results in a response being sent back to the client after the request is processed.

Exercise instructions

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

Reference variable	Windows location	Linux location
<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>\FlatFileOutbound\workspace	
<OUT_DIR>	<LAB_FILES>\ FlatFileOutbound\output	
<FFFILES>	<LAB_FILES>\FFFiles	
<RETRIVE_ARCHIVE>	<LAB_FILES>\ FlatFileOutbound\retrievearchive	
<TEMP>	C:\temp	/tmp

Windows users: When directory locations are passed as parameters to a Java™ program such as EJBdeploy or wsadmin, you must replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\Labfiles62\ with C:/Labfiles62/.

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**".

Part 1: Initialize the workspace and prepare for the lab

This part of the lab, you will start the WebSphere Integration Developer V6.2 with a new workspace and extract the lab files to your local system.

____ 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
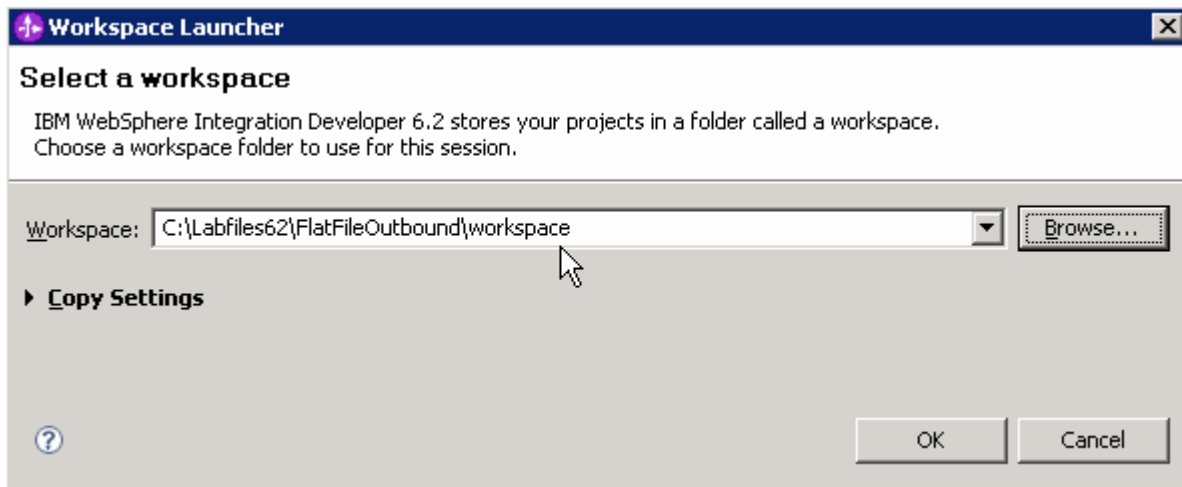
____ 3. Click the button on the right corner to close the Welcome page and proceed with the workbench

Part 2: Review properties

This part of the lab will give you brief description of some the properties which are used in this lab at various stages.

- **New in V6.2 for sequence file**: For each request, the adapter increments the number in the sequence file and the input type takes the sequence number that is currently stored in the sequence file. Sequence numbers are not maintained separately for different input data types.

    For compatibility with sequence files generated with previous versions of the adapter, where sequence numbers were maintained separately for different input data types, the adapter checks for all entries in the file that have the older format (<dirPath>/xyz.txt = 2, where xyz.txt is the file name and 2 is the sequence number to be used when the adapter receives another Create request on the same file). The adapter searches for all such sequence numbers for each input type and uses the highest sequence number as the sequence number for the next input type. The adapter then overwrites the entire file with the new (incremented) sequence number.

    Important: Unless they are part of a cluster, two adapter instances should not access the same sequence file, because this can result in delayed processing of batch requests.

    If the sequence file is deleted manually, the sequences are lost and will start from 1 again. You can also reset the sequence by changing the sequence value in the sequence file.

- **Default target file name**: This value if specified at Managed Connection factory level is used as default to create the new file. You can use this along with Sequence file.

- **chunkFileName:** this is populated during Inbound or on Retrieve operation during outbound and the presence of this indicates that it is a chunked file. This is used for Inbound and Retrieve outbound operations where chunking is enabled.

- **fileContentEncoding:** This encoding is used while writing to the file. If this property is not specified, the RA tries to write without using any specific encoding. You can specify any Java supported encoding set like UTF-8 for this attribute. If the file content is non-English, the corresponding encoding needs to be chosen so the adapter uses the encoding while writing to the file system.

- **includeEndBODelimiter:** This is used during the outbound Create/Append/Overwrite operations. The File content is appended with the value of IncludeEndBODelimiter. For example, if the operation chosen is Append and the specified values for this property is ####, when the BO content is written to the file, the include BO Delimiter content is also appended as part of the BO content at the end of the file.

- **stagingDirectory:** This directory is used only for the create and overwrite operations. The file is written to the staging directory completely and then just moved to the Output directory specified in the directoryPath

- **generateUniqueFile:** During outbound Create operation the adapter creates a unique file when this property is true. When this property is set to true the adapter ignores any value set for file name property.

    The name of the unique file generated by Flat Files adapter will have this format:

    A random number is prefixed by '**ffa**' and with an extension '**.tmp**'. For example, ffa23423.tmp

---

Note: If the **Sequence file** has a value and an output file name is specified, with the 'generateUniqueFile' is set, the 'generateUniqueFile' property takes precedence.

- **createFileIfNotExists:** During Append and Overwrite operations, if the file does not exist, then the adapter creates the file when this property is set to true. If this property is false and file does not exist then the adapter flags an error.

  And while creating file for this condition, if 'generateUniqueFile' is also set to true, then the adapter generates a unique file. At this time the adapter ignores the value present in file name property. If the file to be appended does not exist and this property is set to false, a RecordNotFoundException is thrown to the calling component.

- **splitFunctionClassName:** This value takes a fully qualified class name of the class to be used in order to split the retrieved file during outbound retrieve operation. It takes two values as of now:

  com. ibm.j2ca.utils.filesplit.SplitBySize - a class which splits the file based on file size

  com.ibm.j2ca.utils.filesplit.SplitByDelimiter - a class which splits the file based on delimiter (used to separate BO's in event file)

  The delimiter or file size is given in SplitCriteria.

  If RetrieveContentType is null, then this is automatically set to class name which does splitting based on size.

- **splitCriteria:** This attribute takes different values based on value set in splitFunctionClassName.

  If splitFunctionClassName is set to com. ibm.j2ca.utils.filesplit.SplitByDelimiter, then splitCriteria must contain the delimiter which separates the BO's in the retrieved file.

  If splitFunctionClassName is set to com. ibm.j2ca.utils.filesplit.SplitBySize, then splitCriteria must contain a valid number which represents the size in bytes. If retrieved file size is greater than this value, it is split into chunks of this value and so many chunks are posted. If file size is less than this value the entire event file is posted in one shot. When SplitCriteria=0, chunking is disabled.

- **deleteOnRetrieve:** If this property is set to true, during Retrieve operation, after the file content is retrieved, the file is deleted from the directory on the file system.

- **archiveDirectoryforDeleteOnRetrieve:** If the deleteOnRetrieve property is set to true, the adapter will optionally archive (if this directory is valid) the file to this folder before it is deleted.

Part 3: Pass through scenario

Outbound support can be broadly classified into two flows, one that involves data transformation and another without it (pass-through). The incoming BO can be a content specific BO or a generic Flat Files BO. This part of the lab deals with the pass through:
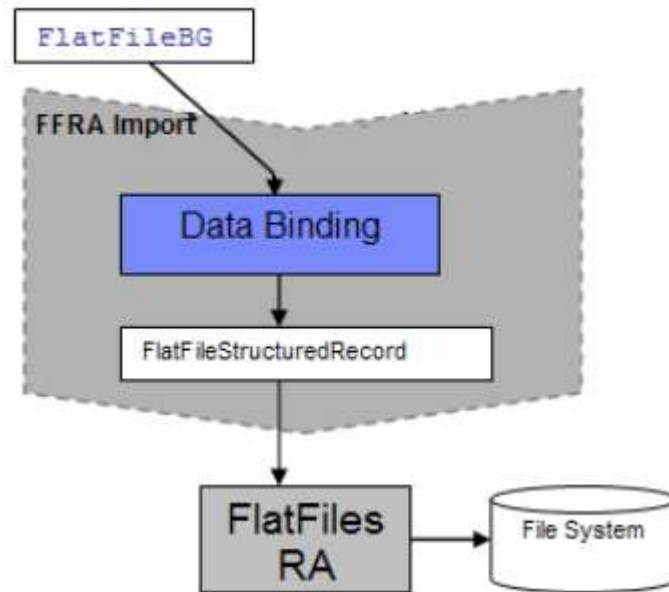
- In the Java EE client, the wrapper data object (FlatFile) is populated for protocol specific information. The actual data object where file content is present (for example: UnstructuredContent) is also set in the wrapper data object. The Java EE client sends the wrapper data object and the outbound operation name as input while making an SCA call.

- Based on the Data Binding (FlatFileBaseDataBinding) configured while running the Enterprise Metadata Discovery, that particular data binding is called and it gets the input data object (FlatFile / FlatFileBG).

- The FlatFileBaseDataBinding invokes the BaseDataBinding. Now, the name of this data object is UnstructuredContent. Based on this condition or the absence of a data handler, the DataBinding does a passThrough. It just instantiates the FlatFileInputStreamRecord, sets the actual content (byte[]) as an InputStream and sets the protocol specific information.

  The output of the outbound operation, if set, is a FlatFileStructuredRecord or FlatFileInputStreamRecord, which is sent back to the Data Binding and mapped to a data object. This data object is sent back to the Java EE client. For create, append and overwrite operations the specific wrapper is returned to the client and for other operations the generic FlatFile wrapper is returned

3.1. Configure pass through using external service wizard

In this part, you will use the new external service feature to create and configure the data binding and operations, which generates the business objects and other artifacts.

____ 1.   Create the module: FFPSOutboundModule

   __ a. From the Business Integration window, right-click and select **New > Module**

   __ b. From the New Module window, enter **FFPSOutboundModule** for the Module Name
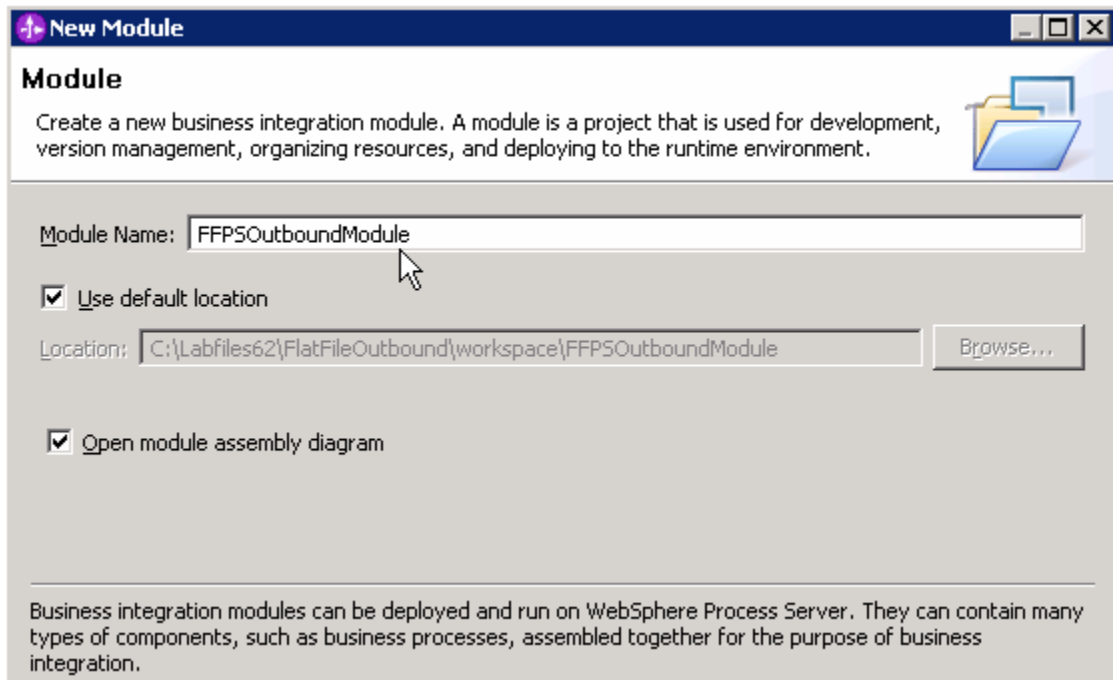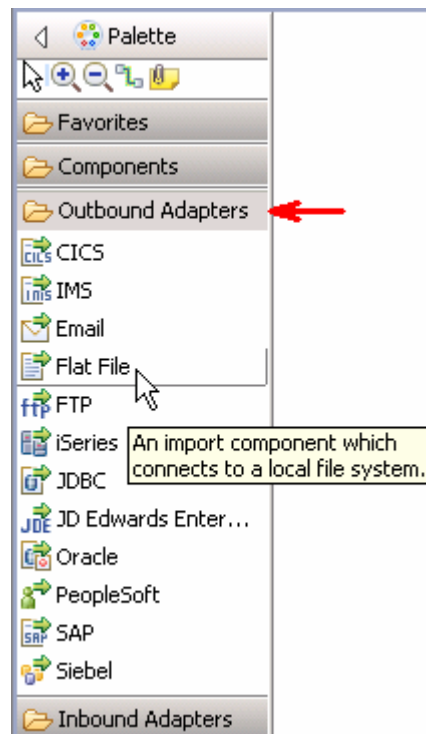

   __ c. Ensure that the box next to **Open module assembly diagram** is checked and then click **Finish**. You will now see a new module, FFPSOutboundModule, created in your Business Integration window

____ 2.   To start External Service from the Palette:
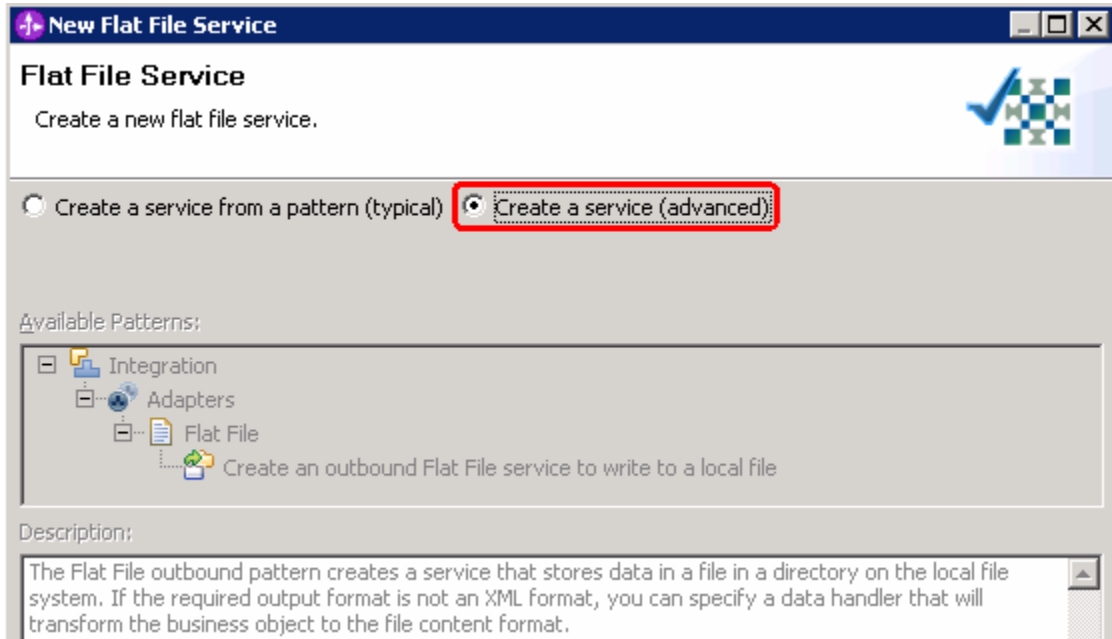
__ a. From the **Palette** on the left side of Assembly Diagram, click **Outbound Adapters**:

____ 3.   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

_____ 4.   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. The wizard provides three connectivity options – Adapters, Registers, and Messaging

Select the radio button next to **Adapters** and click **Next**

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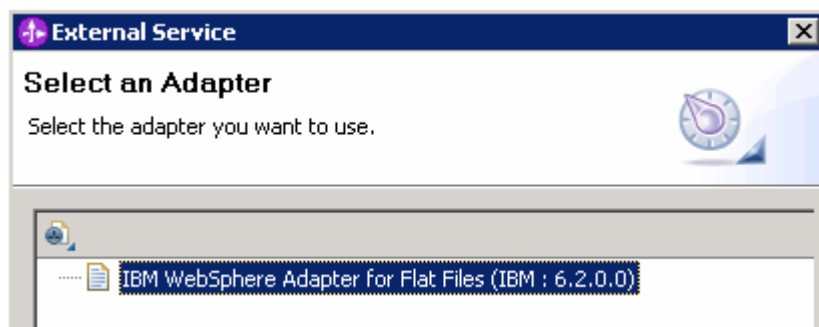

_____ 6.   Adapter Import screen:

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

__ 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 v6.2** is selected

External Service

Adapter Import

Import a resource adapter archive (RAR) from the file system to create a connector project for the adapter.

Archive file: C:\IBM\WID62\ResourceAdapters\FlatFile_6.2.0.0\deploy\CWYFF_FlatFile.rar

Connector project:* CWYFF_FlatFile

Target runtime: WebSphere Process Server v6.2
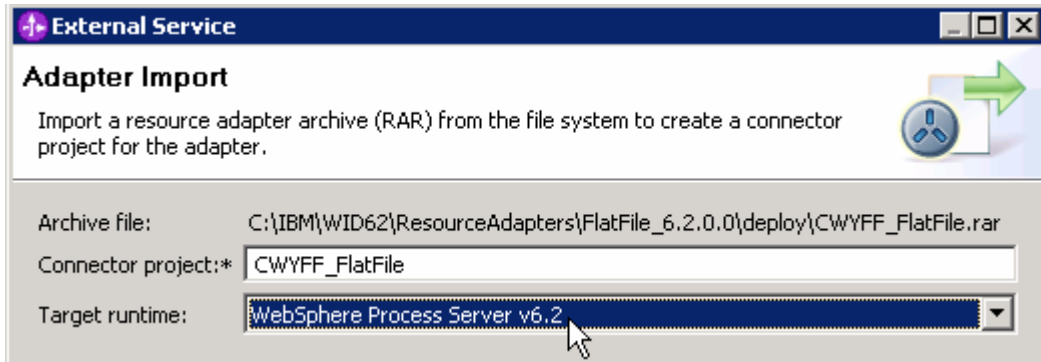
__ d. Click **Next**

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 **Outbound** and click **Next** to proceed to the next step.
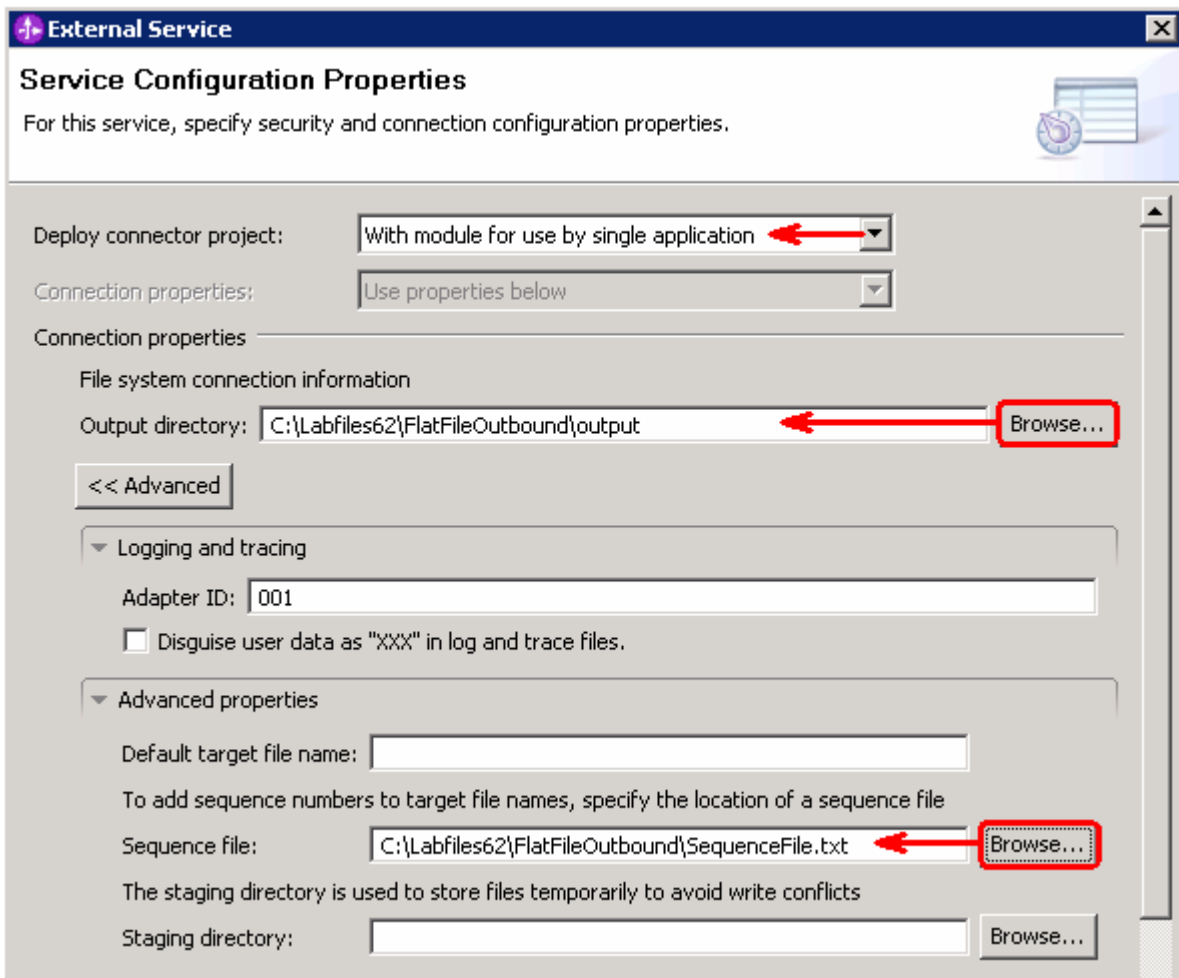
____ 7.   Service Configuration Properties:

__ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected

__ b. Click **Browse…** next to **Output Directory** and select **<OUTPUT_DIR>**

Note: Alternatively, you can also replace the absolute directory path with WebSphere variables for the Event directory, Archive directory. Refer to '**Flat File adapter – Processing COBOL copy book files lab**' for more details on this new feature introduced in V6.2.

____ 8.   Click **Advanced** at the bottom to see the Advanced properties

__ a. **Logging and tracing**: Refer to '**Log and confidential trace lab**' for more details on this new feature

__ b. Under **Advanced properties**, you will find three properties: **Default target file name** and **Sequence file** and Staging directory

__ c. Click **Browse…** next to **Sequence file** and enter any location. For ex: **<LABFILES>\FlatFileOutbound\SequenceFile.txt**

Note: Refer to Review Properties part of this lab for the detailed explanation of this new feature. This file should already be created at the specified location.

____ 9. 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)
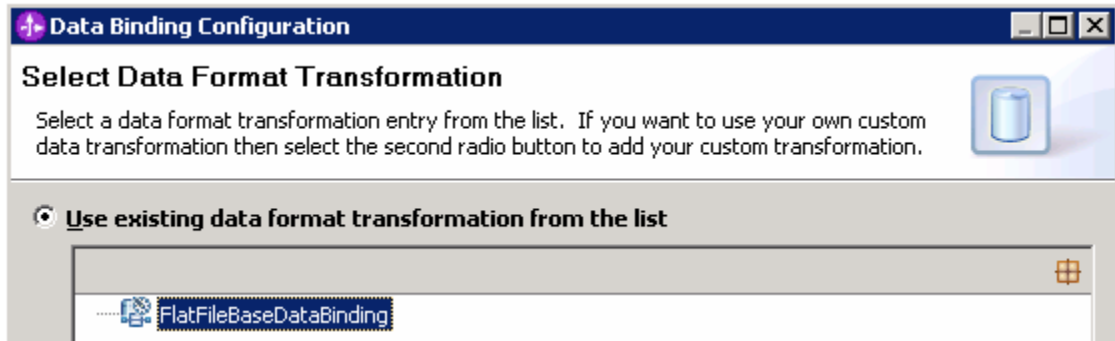
   __ a. From the dropdown menu next to Data format options, select '**Use a data binding configuration for all operations**'
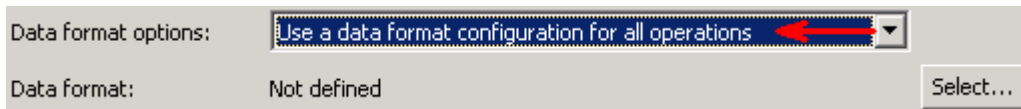
   __ b. Click **Select…** next to **Data format**. A Binding Resource Configuration window is opened.
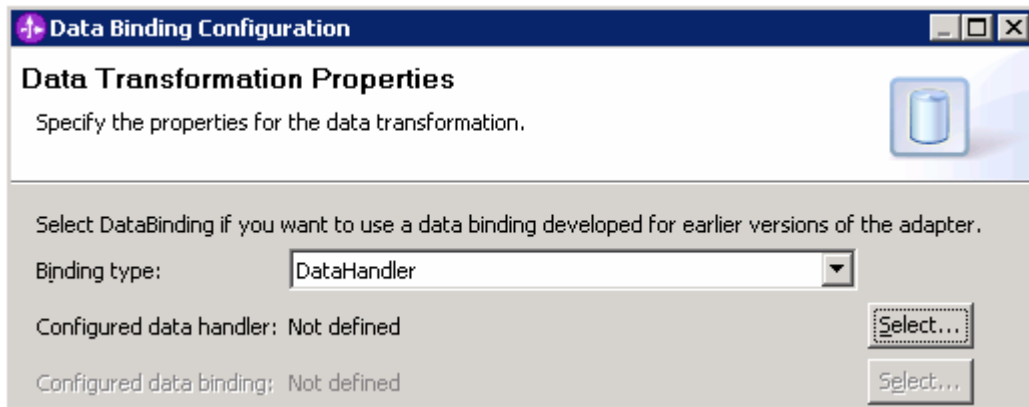
__ c. Select the radio button for **'Use existing data format transformation from the list'** and then select **FlatFileBaseDataBinding**
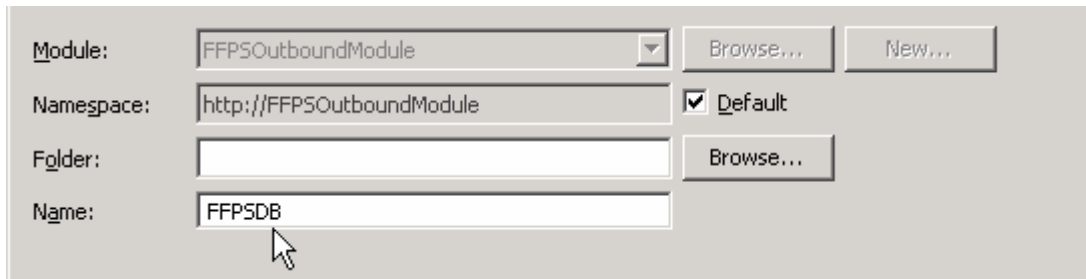
__ d. Click **Next**

Note: Data Handler Configuration: Since you are doing the pass through scenario, you do not need to configure any data handler.

__ e. Click **Next** from the Data Transformation Properties screen

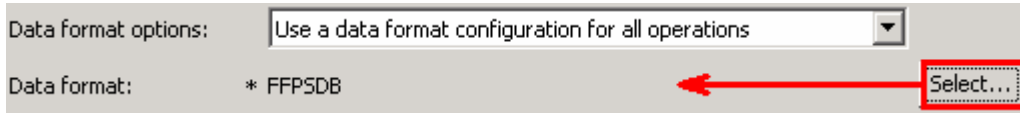__ f. Note that the selected module is **FFPSOutboundModule**

    1) For the **Name**, enter **FFPSDB**

    2) Click **Finish**

WBPMv62_AdapterFFOutboundLab.doc

___ g. Now the **FFPSDB** should be displayed for Data format

Data format options:	Use a data format configuration for all operations ▼	
Data format:	* FFPSDB	← Select...

_____ 10. Check the box next to **Change logging properties for wizard** to view the output location of the log file and the logging level. You can change the logging level using the drop down menu.

    ___ a. Click **Next**

Define Operations: In this screen, you will add the required operations that are used to access functions on the EIS

Note: The precedence of the parameters is as follows: WrapperBO, Interaction Spec, and Managed Connection Factory. The adapter will first search for the parameters passed in the WrapperBO; if it is not available there, it will then subsequently search in the Interaction Spec, and then the Managed Connection Factory instance. **In this lab, for all the operations, you will enter the values at the WrapperBO level in the later part using the WebSphere Integration Developer test client.**
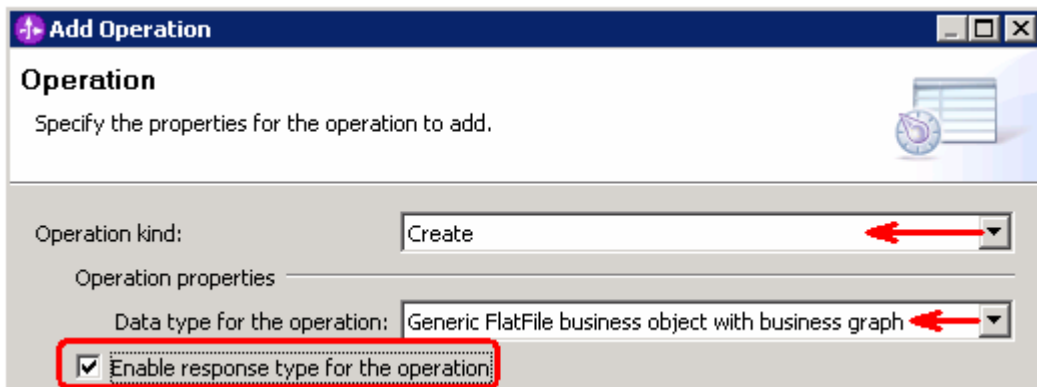

Define Operation: createFFBG

_____ 11. Click **Add…** to open Add Operation window

    ___ a. For **Operation kind**, select **Create** from the drop down list

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

    ___ c. Select the check box for '**Enable response type for the operation**'

    ___ d. Click **Next**

The Data type for input and output are populated based on the selection of the Data type for the operation in the previous step. Since you have chosen Generic FlatFile business object with business graph, the **Input type** is **FlatFileBG** and because you have selected Output required box, the **Output type** is **CreateResponseBG**

Operation name:	* createFlatFile	

Specify the operation input

Input type: → FlatFileBG {http://www.ibm.com/xmlns/prod/websphere/j2ca | Browse... | New...

Data format options: Use suggested data format 'FlatFileBaseDataBinding' ▼

Data format: Not defined | Select...

Specify the operation output

Output type: → CreateResponseBG {http://www.ibm.com/xmlns/prod/websp | Browse... | New...

Data format options: Use suggested data format 'FlatFileBaseDataBinding' ▼

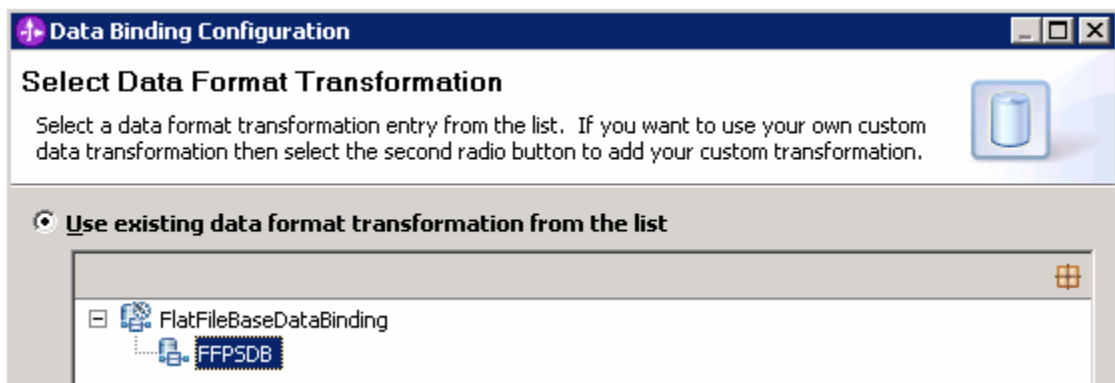Data format: Not defined | Select...

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

Define Data format for **input**:

__ f. For **Data format options**, select **Use a data binding configuration** from the dropdown list

__ g. Click **Select…** next to **Data format**. A Binding Resource Configuration window is opened

__ h. Ensure that the radio button for 'Use existing data format transformation from the list' and then select **FlatFileBaseDataBinding > FFPSDB**

Data Binding Configuration

Select Data Format Transformation

Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.

⦿ **Use existing data format transformation from the list**

⊟ FlatFileBaseDataBinding
　　FFPSDB

__ i. Click **Finish**

Define Data format for **output**:

__ j. Repeat the steps that you did to define the data binding for input and select **FFPSDB**

The Operation screen now should look like this:

__ k. Click **Finish** from the Add Operation window

The operation, createFFBG, will now be displayed under Operations list.
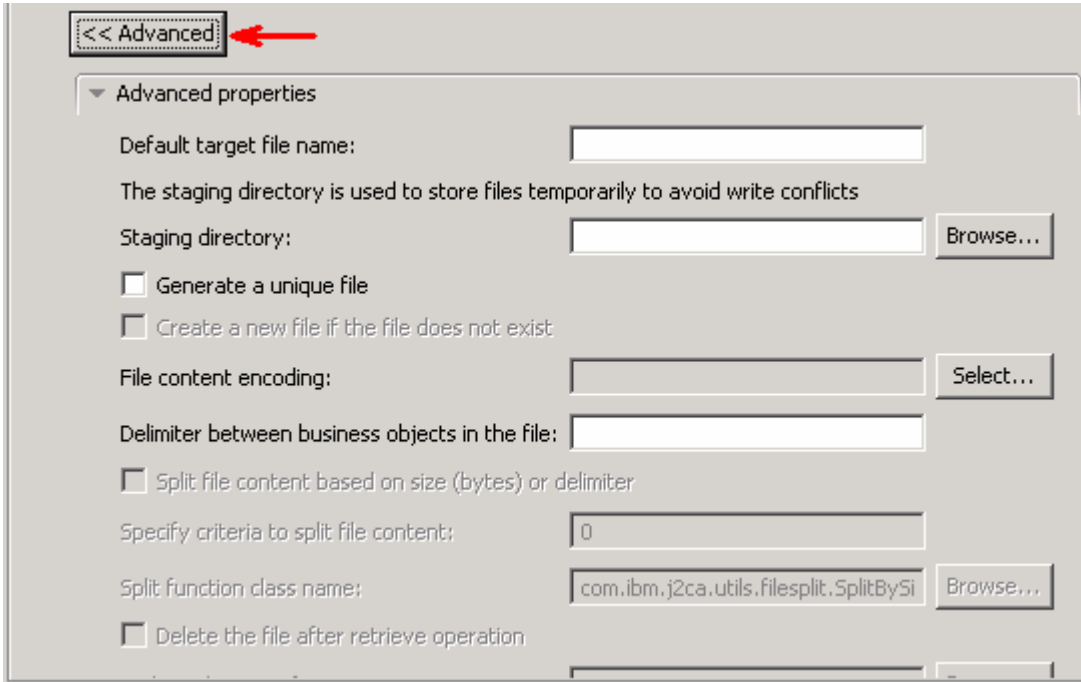
____ 12. You can click **Advanced >>** under 'InteractionSpec properties for createFFBG' to review the properties available at Interaction spec level
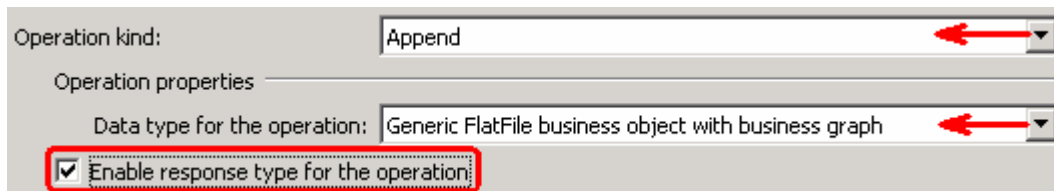

Define Operation: appendFFBG:

____ 13. Click **Add…** to open Add Operation window

___ a. For **Operation kind**, select **Append** from the drop down list

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

___ c. Select the check box for **'Enable response type for the operation'**

___ d. Click **Next**

The Data type for input and output are populated based on the selection of the Data type for the operation in the previous step. Since you have chosen Generic FlatFile business object with business graph, the Input type is **FlatFileBG** and because you have selected Output required box, the **Output** type is **AppendResponseBG**

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

Define Data format for **input**:

__ f. For **Data format options**, select **Use a data binding configuration** from the dropdown list

__ g. Click **Select…** next to **Data format**. A Binding Resource Configuration window is opened.

__ h. Ensure that 'Use existing data transformation from the list', select **FlatFileBaseDataBinding > FFPSDB** and click **Finish**

Define Data format for **output**:

__ i. Repeat the steps you did to select data format for input and select **FFPSDB** for output

The Operation screen now should look like this:

__ j. Click **Finish** from the Add Operation window

The operation, appendFFBG, will now be displayed under Operations list.

_____ 14. You can click **Advanced >>** under 'InteractionSpec properties for appendFFBG' to review the properties available at Interaction spec level
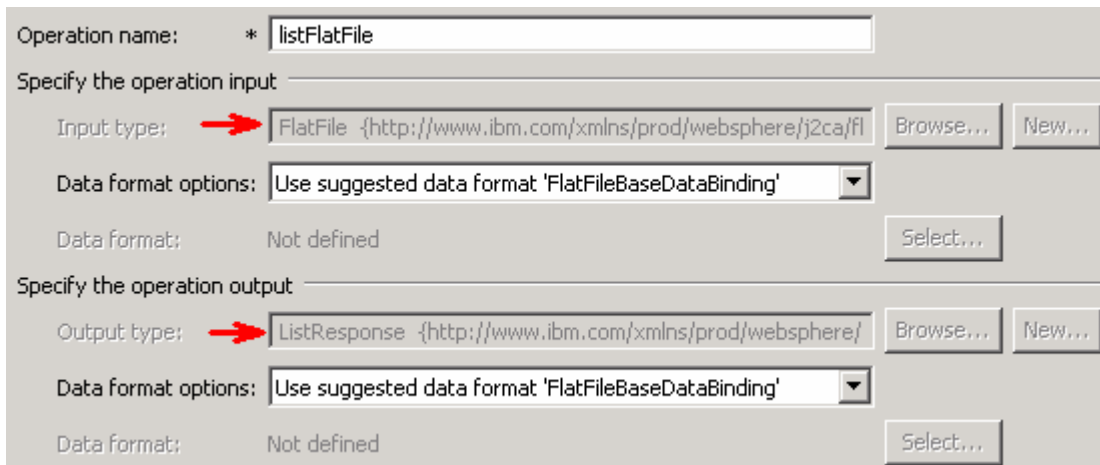

Define Operation: listFF:

_____ 15. Click **Add…** to open Add Operation window

__ a. For **Operation kind**, select **List** 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 check box next to **'Enable response type for the operation'** is selected by default
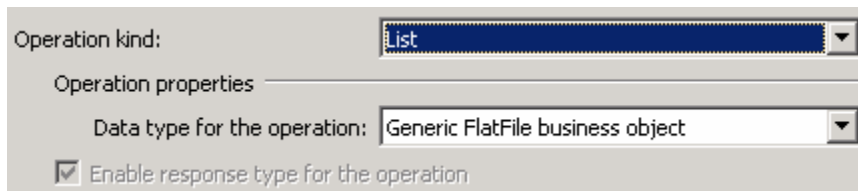
__ d. Click **Next**

The Data type for input and output are populated based on the selection of the Data type for the operation in the previous step. Since you have chosen Generic FlatFile business object, the **Input type** is **FlatFile** and because the Output required is also selected, the **Output type** is **ListResponse**
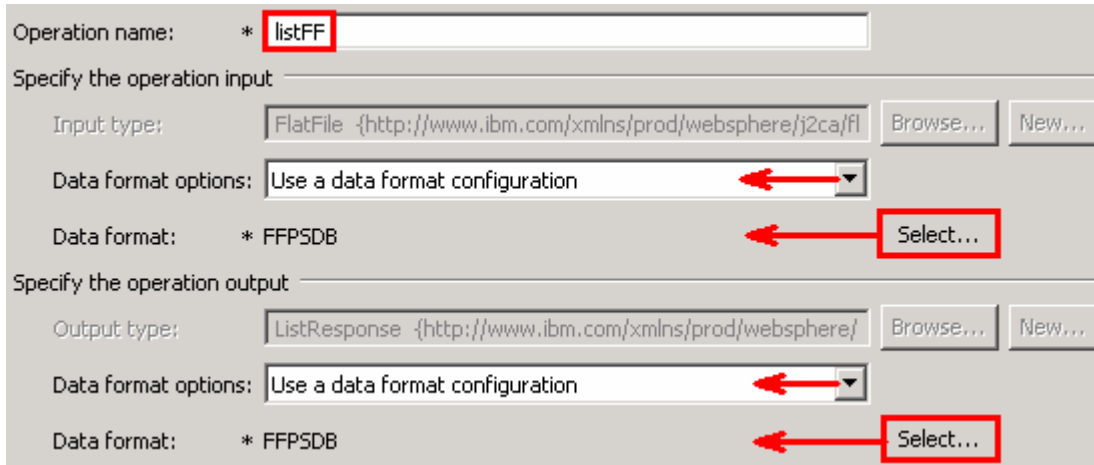

__ e. For Operation name, enter **listFF**

Define Data format for **input** and **output**:

___ f. Repeat the steps you did for Create or Append operation to define the data format and select **FFPSDB** for both **input** and **output**
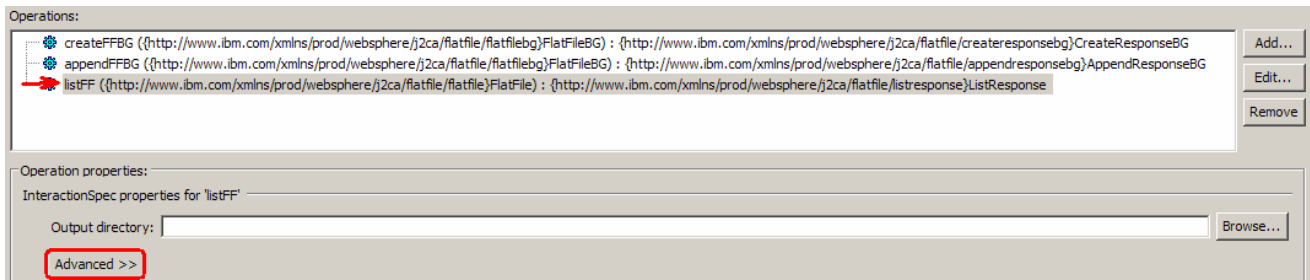
___ g. The Operation screen now should look like this:

___ h. Click **Finish** from the Add Operation window

The operation, **listFF**, will now be displayed under Operations list.

_____ 16.   You can click **Advanced >>** under 'InteractionSpec properties for listFF' to review the properties available at Interaction spec level

Define Operation: retrieveFF:

_____ 17.   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 check box for **'Enable response type for the operation'** is selected by default

___ d. Click **Next**

The Data type for input and output are populated based on the selection of the Data type for the operation in the previous step. Since you have chosen Generic FlatFile business object, the **Input type** is **FlatFile** and because the Output required is also selected, the **Output type** is **RetrieveResponseWrapper**. Also note that the Output type can be modified. But, for the pass through scenario, the output type is going to be the default, RetrieveResponseWrapper.

Operation name:	* retrieveFlatFile

Specify the operation input

Input type: ➡	FlatFile {http://www.ibm.com/xmlns/prod/websphere/j2c	Browse...	New...
Data format options:	Use suggested data format 'FlatFileBaseDataBinding' ▾	
Data format:	Not defined	Select...

Specify the operation output

Output type: ➡ *	RetrieveResponseWrapper {http://www.ibm.com/xmlns/	Browse...	New...
Data format options:	Use data format configuration 'FFPSDB' ▾	
Data format:	Not defined	Select...

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

Define Data format for **input**:

__ f. Repeat the steps you did for Create or Append operation to define the data format and select **FFPSDB** for both **input**

Define Data format for **output**:

__ g. Accept the default selection, **Use data format configuration 'FFPSDB'**, from the drop down list

__ h. The Operation screen now should look like this:

Operation name:	* retrieveFF

Specify the operation input

Input type:	FlatFile {http://www.ibm.com/xmlns/prod/websphere/j2c	Browse...	New...
Data format options:	Use a data format configuration ⬅ ▾	
Data format:	* FFPSDB ⬅	Select...

Specify the operation output

|	
Output type:	* RetrieveResponseWrapper {http://www.ibm.com/xmlns/	Browse...	New...
Data format options:	Use data format configuration 'FFPSDB' ⬅ ▾	
Data format:	Not defined	Select...

__ i. Click **Finish** from the Add Operation window. The operation, retrieveFF, will now be displayed under Operations list.

____ 18. You can click **Advanced >>** under 'InteractionSpec properties for retrieveFF' to review the properties available at Interaction spec level

____ 19. Click **Next** from the Operations window

____ 20. From the Generate Artifacts screen, enter these:

__ a. For **Name**, enter **FFPSOutboundInterface**
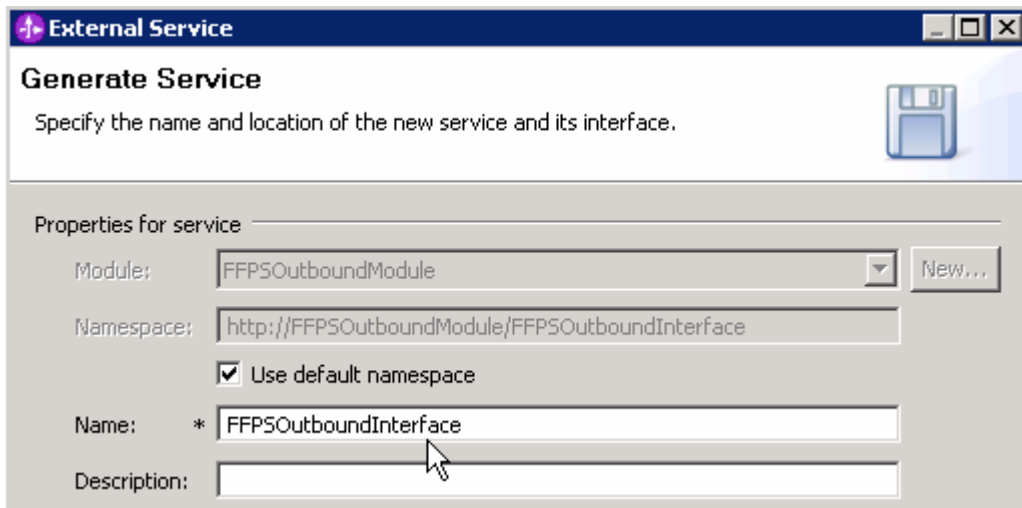
__ b. Click **Finish**

____ 21. You will now see a new import component, **FFPSOutboundInterface** in the assembly diagram of FFPSOutboundModule
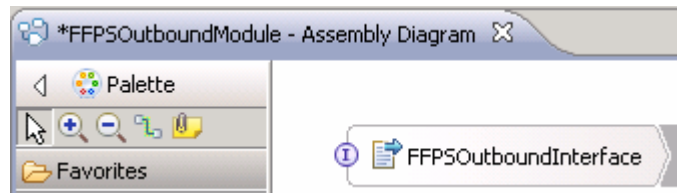
__ a. Save (**Ctrl+S)** your changes to the assembly diagram

____ 22. Review the FFPSOutboundModule: The generated **Data Types**, **Interface**, and the Data binding (**FFPSDB**) under Configured Resources can be found inside FFPSOutboundModule
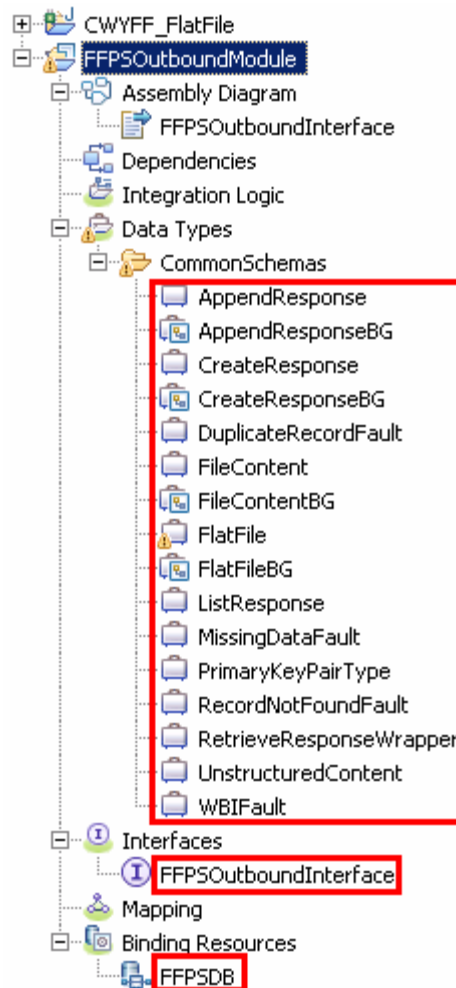
You can open each of these generated artifacts and business objects and review the properties inside.

Review the created methods inside the interface:

__ a. From the Business Integration view, expand FFPSOutboundModule > Interfaces and then double-click **FFPSOutboundInterface** to open it

___ b. You should see these four operations. Note the Input and Output types, which should match the ones that were populated by default while defining each of the operations:

Name	Type
(I) FFPSOutboundInterface ☒	
▼Operations	
Operations and their parameters	

	Name	Type
▼ **createFFBG**		
▷] Input(s)	createFFBGInput	FlatFileBG
(▷ Output(s)	createFFBGOutput	CreateResponseBG
☒ Fault	MISSING_DATA	MissingDataFault
☒ Fault	DUPLICATE_RECORD	DuplicateRecordFault
☒ Fault	RECORD_NOT_FOUND	RecordNotFoundFault
▼ **appendFFBG**		
▷] Input(s)	appendFFBGInput	FlatFileBG
(▷ Output(s)	appendFFBGOutput	AppendResponseBG
☒ Fault	MISSING_DATA	MissingDataFault
☒ Fault	RECORD_NOT_FOUND	RecordNotFoundFault
▼ **listFF**		
▷] Input(s)	listFFInput	FlatFile
(▷ Output(s)	listFFOutput	ListResponse
☒ Fault	MISSING_DATA	MissingDataFault
☒ Fault	RECORD_NOT_FOUND	RecordNotFoundFault
▼ **retrieveFF**		
▷] Input(s)	retrieveFFInput	FlatFile
(▷ Output(s)	retrieveFFOutput	RetrieveResponseWrapper
☒ Fault	MISSING_DATA	MissingDataFault
☒ Fault	RECORD_NOT_FOUND	RecordNotFoundFault

___ c. Close the interface, FFPSOutboundInterface

3.2.   Test pass through 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 Process Server test environment

   __ a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects…** from the pop-up menu

   __ b. In the Add and Remove Projects window, select the **FFPSOutboundModuleApp** project from the Available projects panel

   __ c. Click **Add >** to add it to the Configured projects panel

   __ d. The project is now moved to Configured projects. Click **Finish**

Wait for the project to be published to the server and you can confirm this by seeing this message in the console messages:

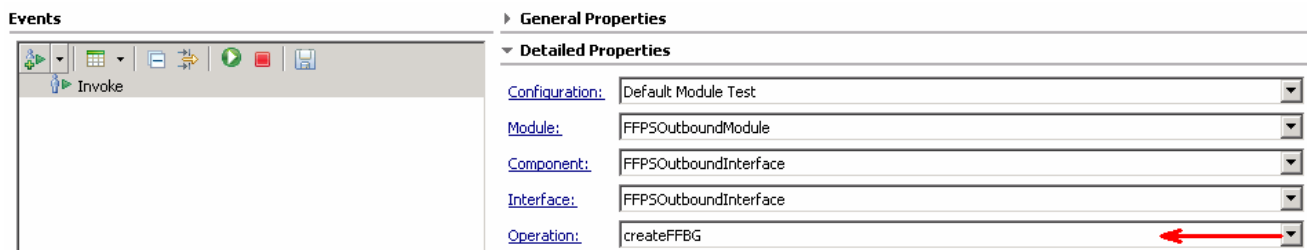> Application started: FFPSOutboundModuleApp

____ 3.   Open the test client for the module

   __ a. From the Business Integration perspective, right-click the **FFPSOutboundModule** and select **Test > Test Module**

   __ b. The **FFPSOutboundModule_Test** window is opened in the Assembly editor

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

- createFFBG

- appendFFBG

- listFF

- retrieveFF

Test Create operation:

____ 4.   Under **Detailed Properties**, for the **Operation** field, select **createFFBG** from the drop down menu

Events		General Properties
		▼ Detailed Properties
▷ Invoke	Configuration:	Default Module Test
	Module:	FFPSOutboundModule
	Component:	FFPSOutboundInterface
	Interface:	FFPSOutboundInterface
	Operation:	createFFBG

____ 5.   Fill out the fields for Initial request parameters:

__ a. For **fileName**, enter any name, for Ex: PassthroughTest.txt

__ b. Under Content. for **AsText**, enter some test message, for Ex: **Testing Create operation for pass through**

Initial request parameters

Name	Type	Value
⊟ createFFBGInput	FlatFileBG	✔
verb	verb<string>	✔ CREATE
⊟ FlatFile	FlatFile	✔
directoryPath	string	✔
fileName	string	✔ PassThroughTest.txt
chunkFileName	string	✔
fileContentEncoding	string	✔
includeEndBODelimiter	string	✔
stagingDirectory	string	✔
chunkNumber	string	✔
generateUniqueFile	boolean	✔ false
createFileIfNotExists	boolean	✔ false
splitFunctionClassName	string	✔
splitCriteria	string	✔
deleteOnRetrieve	boolean	✔ false
archiveDirectoryForDeleteOnRetrieve	string	✔
⊟ Content	UnstructuredContent	✔
ContentType	string	✔
ObjectName	string	✔
AsText	string	✔ Testing Create operation for pass through
AsBinary	hexBinary	✔ 00

__ c. Click **Continue** button under Events

Events

WBPMv62_AdapterFFOutboundLab.doc

__ d. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server v6.2** and click **Finish**
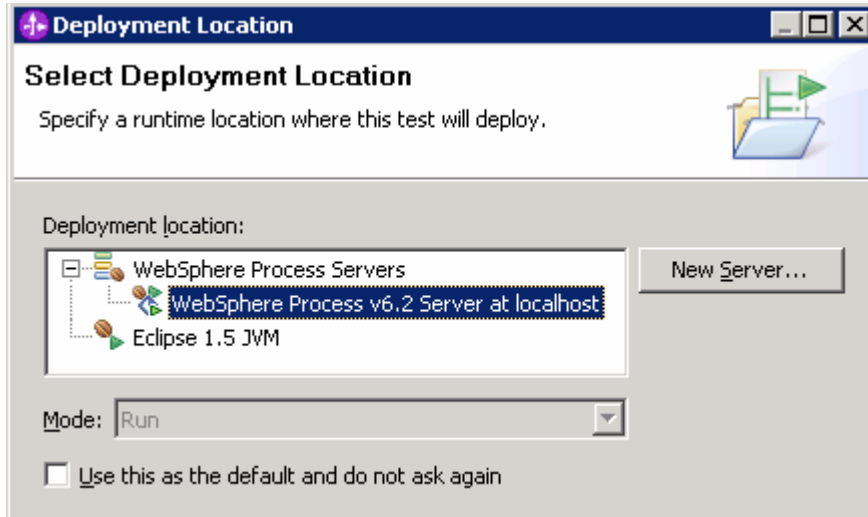
__ e. Provide Administrator User ID and Password

    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
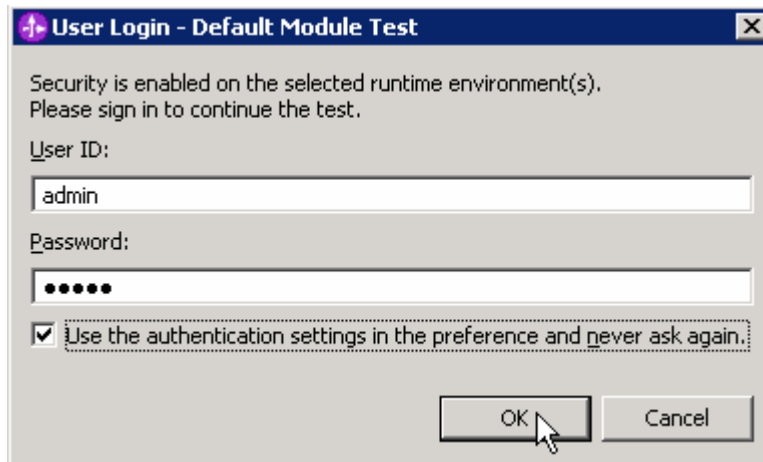
__ f. Click **OK** from Adding/Removing Projects pop-up. Optionally, you can select 'Do not show again' so that you are not required to do this next time when you start the test client
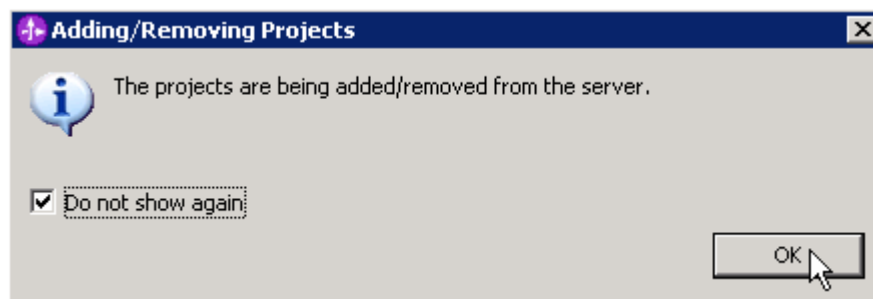
__ g. You should see a window similar to this, which contains the data you just entered in the previous steps:

Note: Since you have specified a Sequence File name while running the External Service, the file name is created with '1' appended to it.
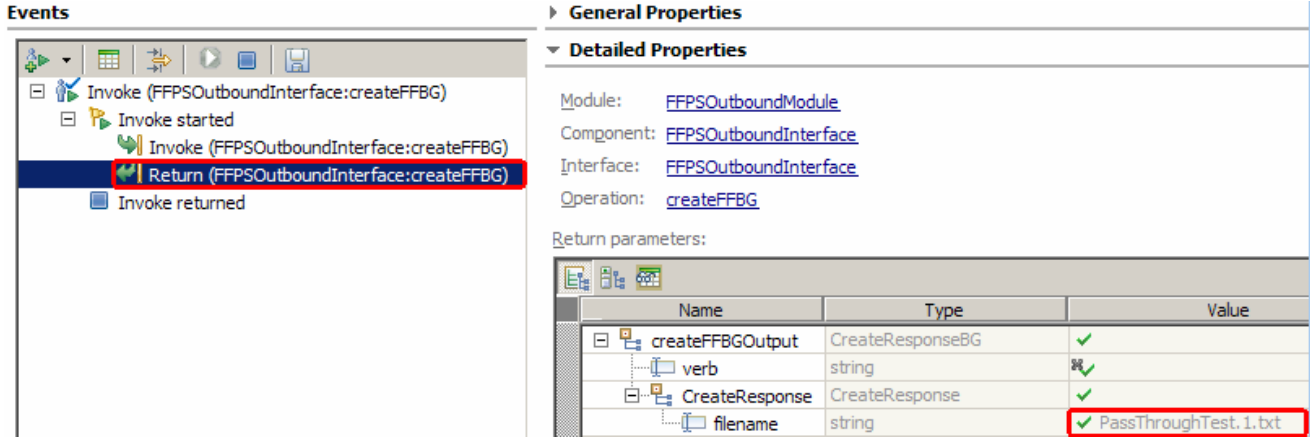
_____ 6.   Verify the created file and its contents

__ a. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

__ b. You will see a new file, **PassthroughTest.1.txt** created under that directory. Double-click it to open the file
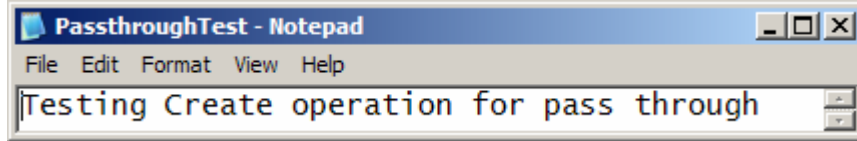
__ c. Now browse to <LAB_FILES>\FlatFileOutbound and open the **SequenceFile.txt** file. You should see a numeric entry in the file. Each time the file is created, the adapter increases the number by one.

Test generateUniqueFile and createFileIfNotExists: In this part you will make use of couple of new features such as generateUniqueFile and createFileIfNotExists.

_____ 7.   Click **Invoke** ( ) under Events to start a new event

_____ 8.   Under **Detailed Properties**, for the **Operation** field, select **appendFFBG** from the drop down menu

Fill out the fields for Initial request parameters:

__ a. For **generateUniqueFile**, select **true** from the drop down list

__ b. For **createFileIfNotExists**, select **true** from the drop down list

Note: You do not need to provide any fileName because createFileIFNotExists is set to true. For append operation, adapter checks this property first and since it is true, it creates a file based on the Boolean value selected for generateUniqueFile. In this case, the value for generateUniqueFile is true, so a unique file is going to be created in <OUT_DIR>.

__ c. Under Content, for **AsText**, enter some test message, for Ex: **Create unique file using append operation**

Initial request parameters

Name	Type	Value
appendFFBGInput	FlatFileBG	✔
verb	verb<string>	✔ CREATE
FlatFile	FlatFile	✔
directoryPath	string	✔
fileName	string	✔
chunkFileName	string	✔
fileContentEncoding	string	✔
includeEndBODelimiter	string	✔
stagingDirectory	string	✔
chunkNumber	string	✔
generateUniqueFile	boolean	✔ true
createFileIfNotExists	boolean	✔ true
splitFunctionClassName	string	✔
splitCriteria	string	✔
deleteOnRetrieve	boolean	✔ false
archiveDirectoryForDeleteOnRetrieve	string	✔
Content	UnstructuredContent	✔
ContentType	string	✔
ObjectName	string	✔
AsText	string	✔ Create unique file using append operation
AsBinary	hexBinary	✔ 00

__ d. Click **Continue** button under Events

__ e. You should see a window similar to this, which contains the data you just entered in the previous steps:

Events

- Invoke (FFPSOutboundInterface:createFFBG)
  - Invoke started
    - Invoke (FFPSOutboundInterface:createFFBG)
    - Return (FFPSOutboundInterface:createFFBG)
  - Invoke returned
- Invoke (FFPSOutboundInterface:appendFFBG)
  - Invoke started
    - Invoke (FFPSOutboundInterface:appendFFBG)
    - Return (FFPSOutboundInterface:appendFFBG)
  - Invoke returned

▶ **General Properties**

▼ **Detailed Properties**

Module: FFPSOutboundModule

Component: FFPSOutboundInterface

Interface: FFPSOutboundInterface

Operation: appendFFBG

Return parameters:

| Name	Type	
appendFFBGOutput	AppendResponseBG	✔
verb	string	✖✔
AppendResponse	AppendResponse	✔
filename	string	✔ ffa44216.tmp

____ 9.   Verify the created file and its contents

__ a. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

__ b. You will see a new file, **ffa44216.tmp** created under that directory. Open the file and observe the content.

Note: Because you set **generateUniqueFile** to true, a file with the name as random number prefixed by '**ffa**' and with an extension '**.tmp**' is created. So, the generated file name is going to be different for you.

```
Create unique file using append operation
```
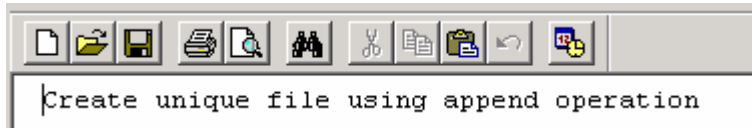
Test Append operation:

____ 10. Click **Invoke** ( ) under Events to start a new event

____ 11. Under **Detailed Properties**, for the **Operation** field, select **appendFFBG** from the drop down menu

Fill out the fields for Initial request parameters:

__ a. For fileName, enter the name of the file you created in the previous test for append. For example ffa44216.tmp (The file should already exist as you are not going to use createFileIfNotExists for this testing)

__ b. Set both **generateUniqueFile** and **createFileIfNotExists** to **false**

__ c. Under **Content**, for **AsText**, enter some test message, for Ex: **Appended content**

__ d. Click **Continue** button under Events

__ e. You should see a window similar to this, which contains the data you just entered in the previous steps:

____ 12.   Verify the created file and its contents

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

__ a. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

__ b. Open the file you specified and observe the highlighted content, which is appended to the original content
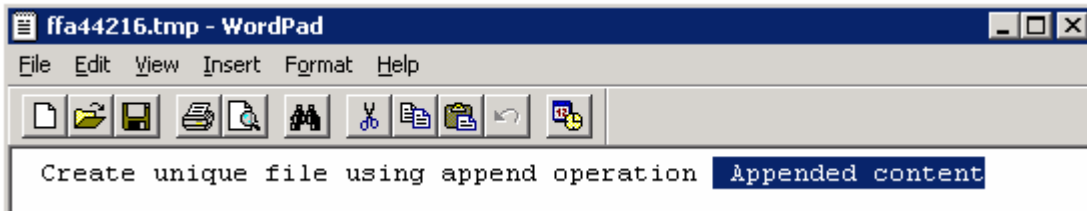
Test List operation:

____ 13.   Click **Invoke** ( ) under Events to start a new event

____ 14.   Under **Detailed Properties**, for the **Operation** field, select **listFF** from the drop down menu
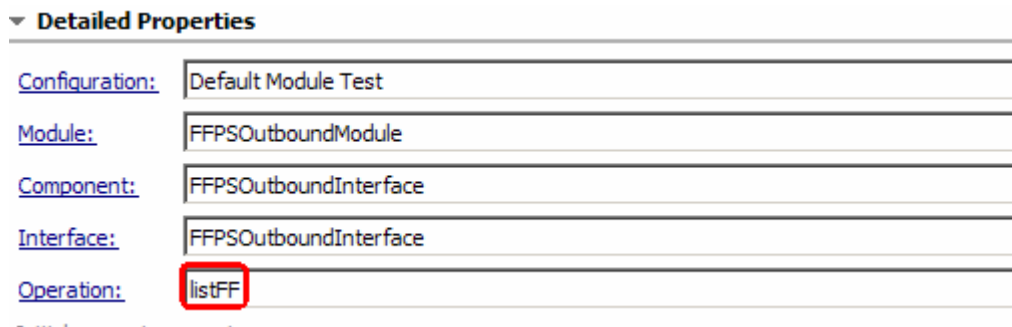
____ 15.   Fill out the fields for Initial request parameters:

__ a. You can leave all the other fields empty. The directoryPath, <OUT_DIR>, is already provided at Managed Connection Properties level during the External Service Wizard

__ b. Click **Continue** button under Events

__ c. Test client will return the list of files under the <OUT_DIR>. You should see the two listOfFileNames: **ffa44216.tmp**, **PassthroughTest.1.txt** since those are created in the previous steps.

Test Retrieve operation:

____ 16.   Click **Invoke** ( ) under Events to start a new event

____ 17.   Under **Detailed Properties**, for the **Operation** field, select **retrieveFF** from the drop down menu

Fill out the fields for Initial request parameters:

__ a. For **fileName**, enter the name of the file used for append operation. Ex: **ffa44216.tmp** (The file name should already exist for retrieve operation. ffa44216.tmp was created in the previous tests)

__ b. For **deleteonRetrieve**, change it to **true** from the drop down list

__ c. For **archiveDirectoryForDeleteOnRetrieve**, enter the value **<RETRIEVE_ACHIVE>**

Initial request parameters

Name	Type	Value
⊟ 🔡 retrieveFFInput	FlatFile	✔
⬚ directoryPath	string	✔
⬚ fileName	string	✔ ffa44216.tmp
⬚ chunkFileName	string	✔
⬚ fileContentEncoding	string	✔
⬚ includeEndBODelimiter	string	✔
⬚ stagingDirectory	string	✔
⬚ chunkNumber	string	✔
⬚ generateUniqueFile	boolean	✔ false
⬚ createFileIfNotExists	boolean	✔ false
⬚ splitFunctionClassName	string	✔
⬚ splitCriteria	string	✔
⬚ deleteOnRetrieve	boolean	✔ true
⬚ archiveDirectoryForDeleteOnRetrieve	string	✔ C:\Labfiles62\FlatFileOutbound\retrievearchive
⊟ 🔡 Content	UnstructuredContent	✔
⬚ ContentType	string	✔
⬚ ObjectName	string	✔
⬚ AsText	string	✔
⬚ AsBinary	hexBinary	✔ 00

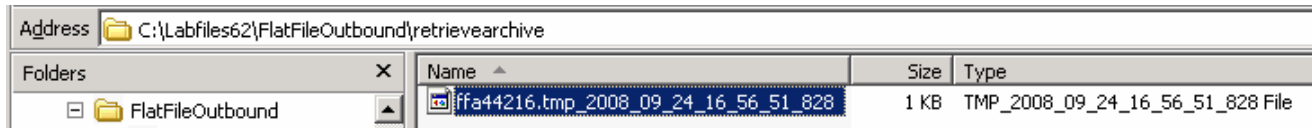__ d. Click **Continue** button under Events

____ 18. Verify the results

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

Name	Type	Value
⊟ 🔡 retrieveFFOutput	RetrieveResponse...	✔
⊟ [◼] Content	anyType[]	60
⊟ 🔡 Content[0]	FileContent	✔
⬚ fileName	string	✔ ffa44216.tmp
⊟ 🔡 fileContent	UnstructuredContent	✔
⬚ ContentType	string	✔ **null**
⬚ ObjectName	string	✔ http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile
⬚ AsText	string	✔ Create unique file using append operation Appended content
⬚ AsBinary	hexBinary	✖✔

__ b. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**. The file ffa44216.tmp is deleted from the list

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

__ c. From the windows explorer, browse to the subdirectory **<RETRIVE_ARCHIVE>** and you will see
a new file created under that directory.

Address	C:\Labfiles62\FlatFileOutbound\retrievearchive		
Folders ✕	Name ▲	Size	Type
⊟ 📁 FlatFileOutbound ▲	📄 ffa44216.tmp_2008_09_24_16_56_51_828	1 KB	TMP_2008_09_24_16_56_51_828 File

__ d. Optionally, double-click it to open the file. Observe the contents of the file

WBPMv62_AdapterFFOutboundLab.doc

3.3.   Restore server configuration

_____ 1.   Close the **FFPSOutboundModule_Test** window and click **No** for the Save Resources window
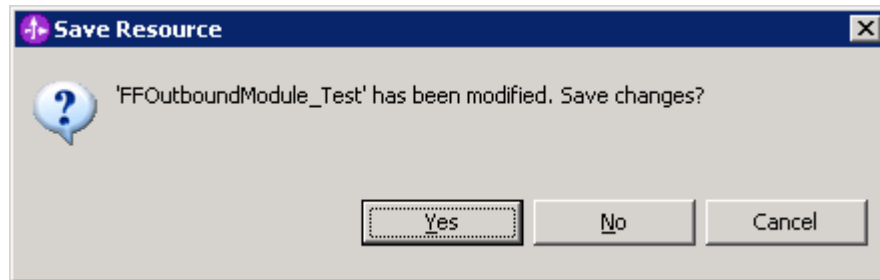
_____ 2.   Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects…** from the pop-up menu

_____ 3.   Select **FFPSOutboundModuleApp** under Configured projects and click **< Remove**

_____ 4.   Click **Finish** after you see the application moved to Available projects. Wait until the application is unpublished

Part 4: Content specific (non-pass through) scenario

Of the two outbound flows, you have just tested the pass-through which does not involve data transformation. In this part of the lab you will perform the non-pass through that uses the data transformation:
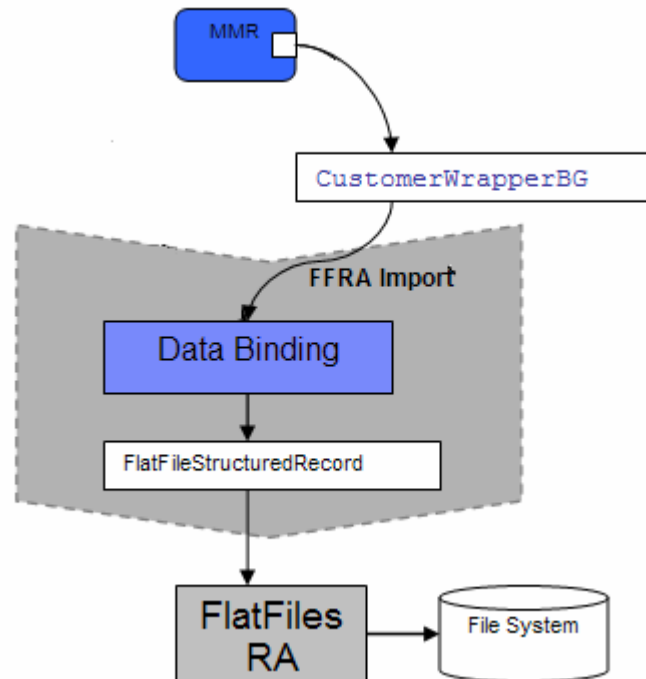
- In the Java EE client, the content-specific wrapper data object (CustomerWrapper) is populated for protocol specific information. If you have specified the actual data object (for example: Customer) as the input type, the protocol specific information is picked up from either the interaction spec or the MCF. The Java EE client sends this wrapper data object and the outbound operation name as input by making an SCA call.

- Based on the Data Binding (FlatFileBaseDataBinding) configured while running the Enterprise Metadata Discovery, that particular data binding is called and it gets the WrapperBG data object.

- The FlatFileBaseDataBinding invokes the BaseDatabinding, which checks the value specified for the data handler property in the data binding properties. Based on the data handler set in the data binding properties, that particular content-specific data handler is invoked.

- The output of the FlatFileBaseDataBinding is a FlatFileInputStreamRecord. The output of the outbound operation is a FlatFileStructuredRecord which is sent back to the Data Binding and mapped to a data object. This data object is sent back to the Java EE client. For create, append and overwrite operations the specific content-specific object is returned to the client and for other operations the generic FlatFile object is returned.

4.1. Configure content specific (non-pass through) scenario using the external service wizard

In this part, you will use the new WebSphere Integration Developer feature, External Service, to create/configure the Data Binding, **Data Handler**, and Operations, which generates the business objects and other artifacts.

_____ 1.   Create the module: FFCustomOutboundModule

 __ a. From the Business Integration window, right-click and select **New > Module**

 __ b. From the New Module window, enter **FFCustomOutboundModule** 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, FFCustomOutboundModule, created in your Business Integration window

_____ 2.   Import required business objects

 __ a. Expand FFCustomOutboundModule (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 ear files under that folder on the right side.

 __ d. Select the box next to **Customer.xsd**

 __ e. Ensure that the **FFCustomOutboundModule** is selected for Into folder

 __ f. Click **Finish** from the Import window

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

_____ 3.   Review the imported business objects:

 __ a. Expand **FFCustomOutboundModule > Data Types** and you will now see a new data type **Customer** under it.

__ b. Double-click **Customer** to view its fields:
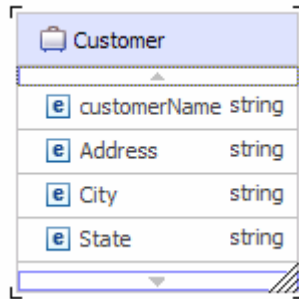
____ 4.   After reviewing, close the Customer business object from the Assembly editor

____ 5.   To start External Service from the Palette:

__ a. From the **Palette** on the left side of Assembly Diagram, click **Outbound Adapters**

____ 6.   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

____ 7.   From the Flat File Service screen, select **Create a service (advanced)**
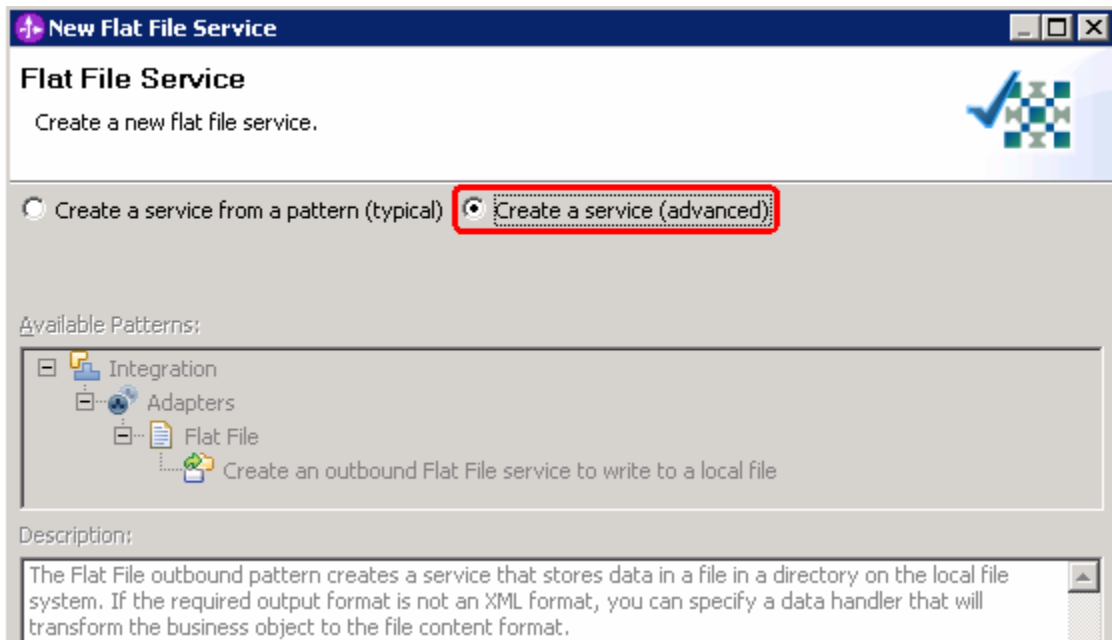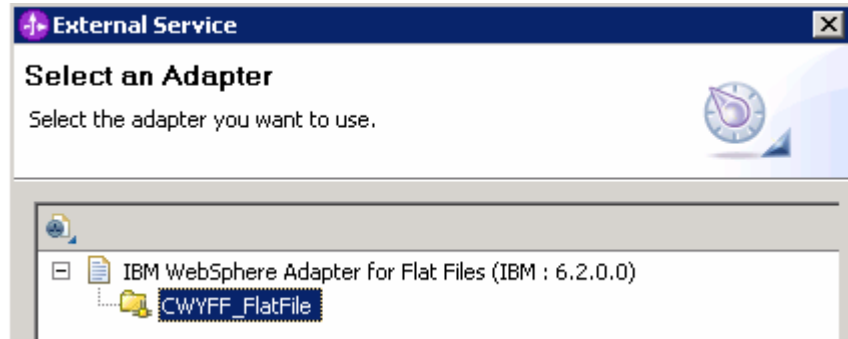

__ a. Click **Next**

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. The wizard provides three
connectivity options – Adapters, Registers, and Messaging.

Select the radio button next to **Adapters** and click **Next**

____ 8. 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**

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 **Outbound** and click **Next** to proceed to the next step.
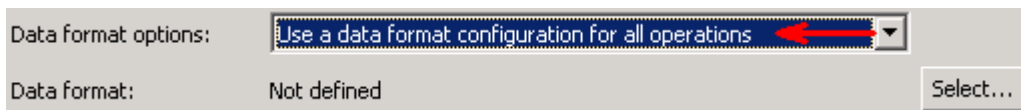

____ 9. Service Configuration Properties:

__ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected

__ b. Click **Browse…** next to **Output Directory** and select **<OUTPUT_DIR>**

__ c. Click **Advanced** at the bottom to see the Advanced and Bidi properties
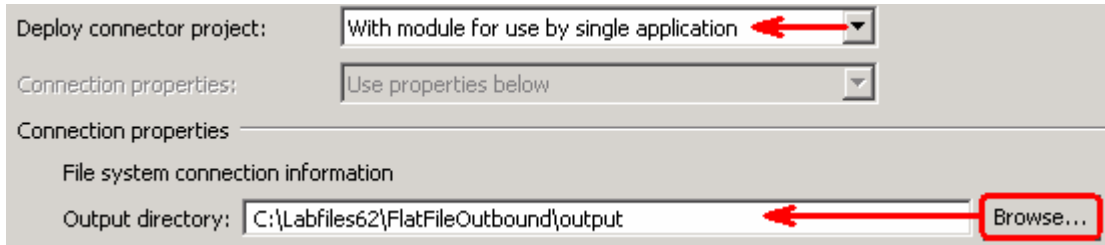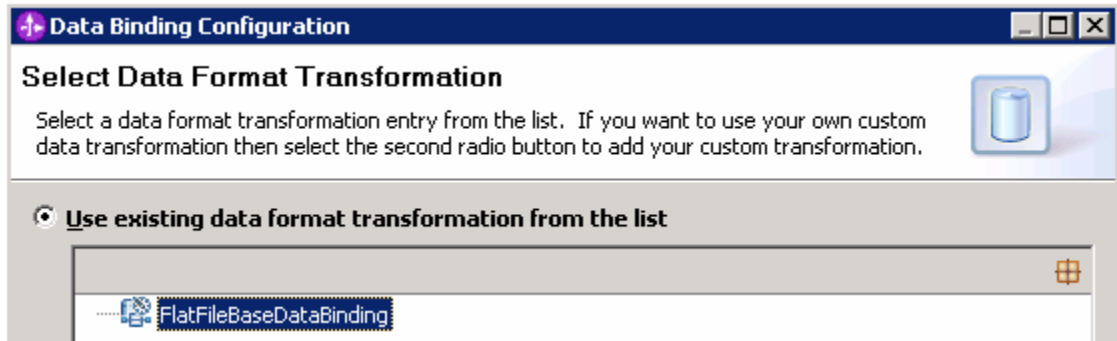
Data binding and Data handler configuration:

____ 10. 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

WBPMv62_AdapterFFOutboundLab.doc

___ c. Select the radio button for '**Use existing data format transformation from the list**' and then select **FlatFileBaseDataBinding**
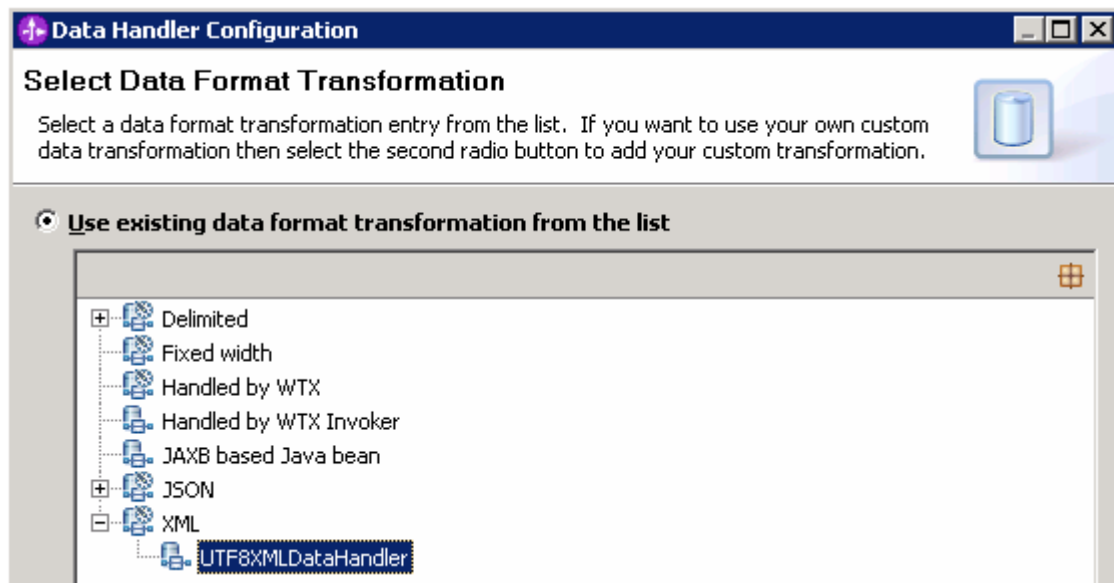

___ d. Click **Next**

_____ 11.   From Data Transformation Properties screen, click **Select…** next to **Configured data handler**. A Binding Resource Configuration window is opened for you to define the data handler

___ a. Select the radio button for '**Use existing data format transformation from the list**' and then select **XML > UTF8XMLDataHandler**
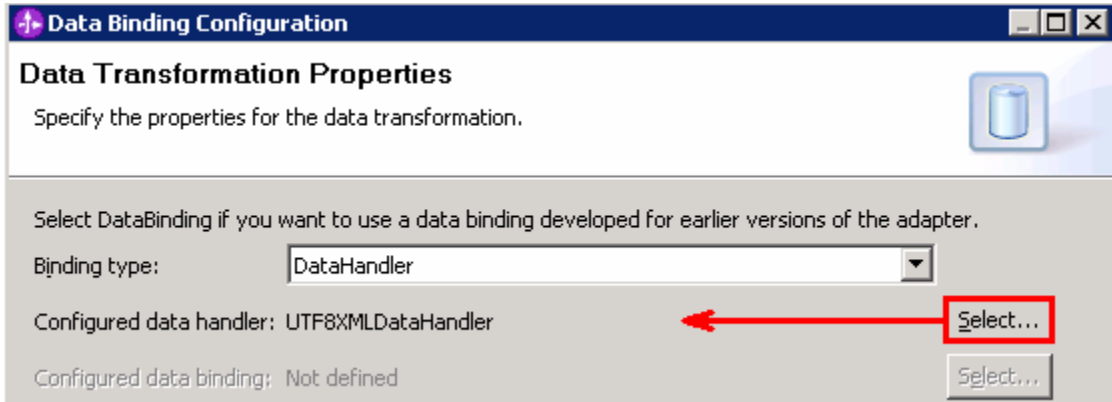
Note: UTF8XMLDataHandler listed under XML is the predefined data handler with UTF-8 as the encoding. You can also select XML and then select the encoding of your choice in the next screen to define a data handler of your choice.

___ b. Click **Finish**

__ c. Back to 'Data Transformation Properties' screen, and the Configured data handler, **UT8XMLDatahandler** is displayed

__ d. Click **Next**

__ e. Ensure that the module selected is FFCustomOutboundModule and enter **CustomDB** for the Name of the data binding
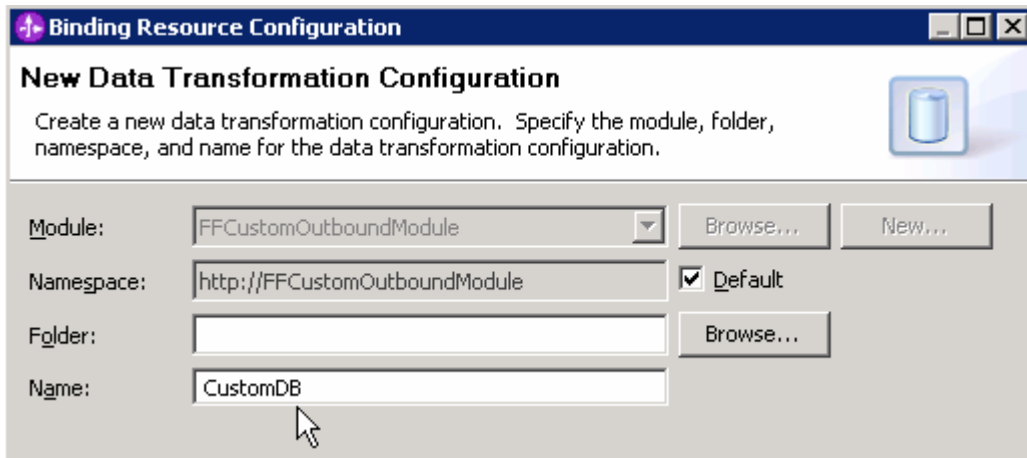
__ f. Click **Finish**

__ g. Now the **CustomDB** should be displayed for **Data format**

____ 12. 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 Operations: In this screen, you will add the required operations that are used to access functions on the EIS

Note: The precedence of the parameters is as follows: WrapperBO, Interaction Spec, and Managed Connection Factory. The adapter will first search for the parameters passed in the WrapperBO; if it is not available there, it will then subsequently search in the Interaction Spec, and then the Managed Connection Factory instance. **In this lab, for all the operations, you will enter the values at the WrapperBO level in the later part using the WebSphere Integration Developer test client.**

Define Operation: createCustomFile:

_____ 13. Click **Add…** to open Add Operation window

___ a. For **Operation kind**, select **Create** from the drop down list

___ b. For **Data type for operation**, select **User defined type** from the drop down list

___ c. Select the check box next to **Enable response type for the operation**
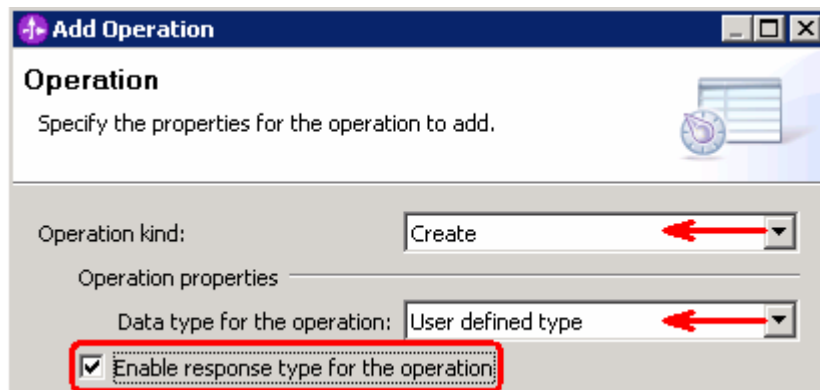
___ d. Click **Next**

The Data type for input and output are populated based on the selection of the Data type for the operation in the previous step. Since you have chosen User defined type, the **Input type** is **blank** and because you have selected Output required box, the **Output type** is **CreateResponse**

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

__ f. Define Input type:

> 1) Under **Specify the operation input**, click **New…** next to **Input type** to open a New Business Object window
>
> 2) Ensure that the Module selected is **FFCustomOutboundModule** and click **Next**
>
> 3) From this window, Click **Browse…** next to Data type
>
> 4) From the Data Type Selection window, select **Customer** under Matching data types:

> 5) Click **OK**

__ g. From the Business Object window, **check** the box next to **Generate business graph for each business object**

__ h. **Do not** check the box for '**Generate retrieve container to retrieve multiple business objects**'

Note: The 'Generate retrieve container to retrieve multiple business objects' is used only during outbound retrieve operation.

| Data type: | * Customer {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customer} | Browse... | New... |

☑ Generate business graph for each business object

Namespace for generated business graph and container object.

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

☐ Generate retrieve container to retrieve multiple business objects.

___ i. Click **Finish**

___ j. In the Add Operation window, under Specify the operation input, you will see the Input type **CustomerWrapperBG** (because you have selected to have business graph (BG) generated)

| Operation name: | * createCustomFile |

Specify the operation input

| Input type: | * CustomerWrapperBG {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customerwrapperbg} | Browse... | New... |

Define Data format for **input**:

___ k. Accept the default Data format options selection, **Use data format configuration 'CustomDB'**

Next define the Data format for **output**:

___ l. For **Data format options**, select **Use a data binding configuration** from the dropdown list

___ m. Click **Select…** next to **Data format**. A Binding Resource Configuration window is opened

___ n. Ensure that the radio button for 'Use existing data format transformation from the list' and then select **FlatFileBaseDataBinding > FixedWidthDB**

___ o. Click **Finish**

___ p. The Operation screen now should look like this:

| Operation name: | * createCustomFile |

Specify the operation input

Input type:	* CustomerWrapperBG {http://www.ibm.com/xmlns/prod,	Browse...	New...
Data format options:	Use data format configuration 'CustomDB'	
Data format:	Not defined	Select...

Specify the operation output

| Output type: | CreateResponse {http://www.ibm.com/xmlns/prod/web | Browse... | New... |
| Data format options: | Use a data format configuration | |
| Data format: | * CustomDB | Select... |

___ q. Click **Finish** from the Add Operation window

The operation, createCustomFile, will now be displayed under Operations list.

__ r. You can click **Advanced >>** under 'InteractionSpec properties for createCustomFile' to review the properties available at Interaction spec level
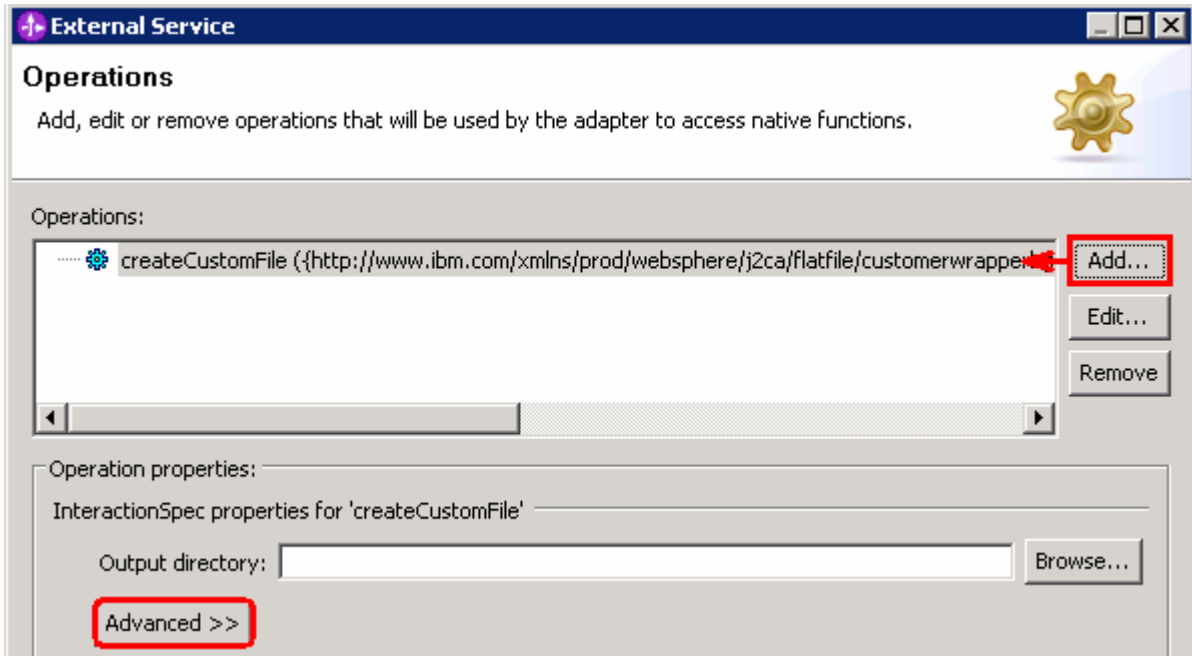
Add Operation: appendCustomFile:

_____ 14. Click **Add…** to open Add Operation window

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

__ b. For **Data type for operation**, select **User defined type** from the drop down list

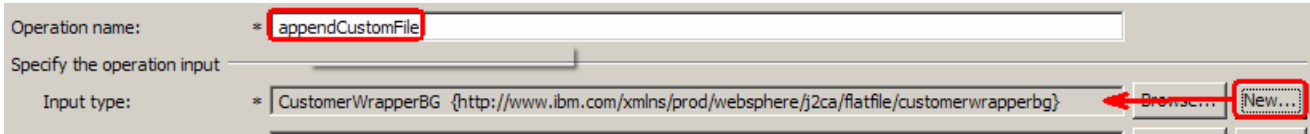__ c. Select the check box next to **Enable response type for the operation**
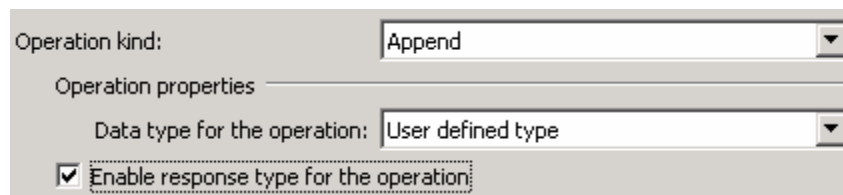
You are back to Operation window and because you chose the User defined data type, the Input type and Output type is blank and because you have selected Output required box, the Output type is AppendResponse.

__ d. Click **Next**

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

__ f. Define Data type for input:

1) Under **Specify the operation input**, click **Browse…** next to **Input type**

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

2) From the Data Type Selection window, select **CustomerWrapperBG** under Matching data types and click **OK**

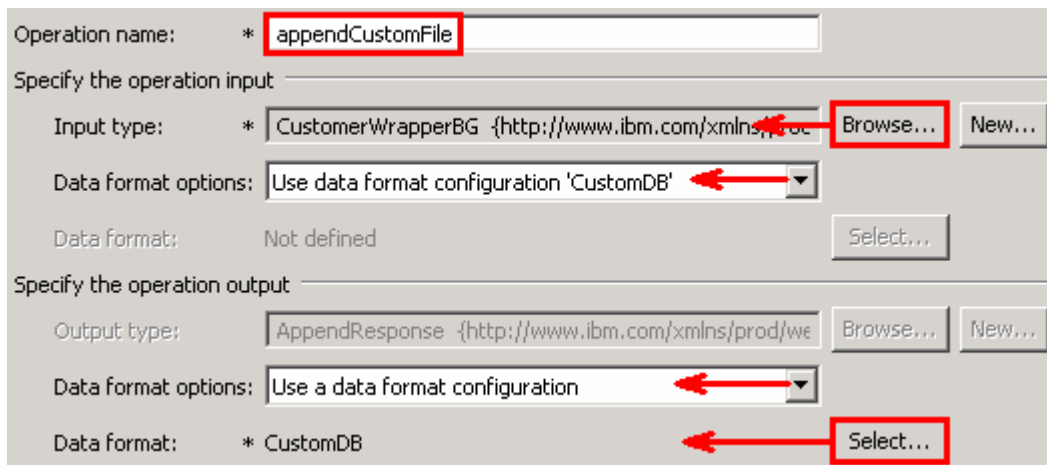|	
Operation name:	* appendCustomFile
Specify the operation input	
Input type:	* CustomerWrapperBG {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customerwrapperbg}

Define Data format for **input**:

__ g. Accept the default Data format options selection, **Use data format configuration 'CustomDB'**

Define Data format for **output**:

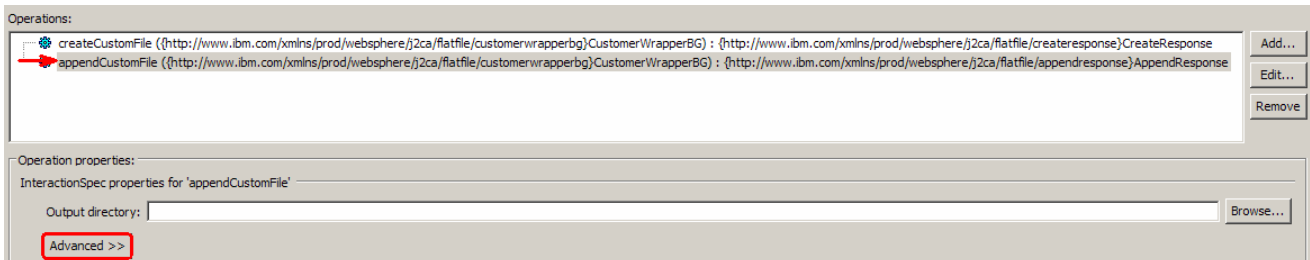__ h. Follow the instructions of defining data format for createCustomFile output and define **CustomDB**

__ i. The Operation screen now should look like this:

Operation name:	* appendCustomFile
Specify the operation input	
Input type:	* CustomerWrapperBG {http://www.ibm.com/xmlns/...
Data format options:	Use data format configuration 'CustomDB'
Data format:	Not defined
Specify the operation output	
Output type:	AppendResponse {http://www.ibm.com/xmlns/prod/we
Data format options:	Use a data format configuration
Data format:	* CustomDB

__ j. Click **Finish** from Add Operation window

The operation, appendCustomFile, will now be displayed under Operations list.

____ 15.   You can click **Advanced >>** under 'InteractionSpec properties for appendCustomFile' to review the properties available at Interaction spec level

Operations:
- createCustomFile ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customerwrapperbg}CustomerWrapperBG) : {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/createresponse}CreateResponse
- appendCustomFile ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customerwrapperbg}CustomerWrapperBG) : {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/appendresponse}AppendResponse

Operation properties:
InteractionSpec properties for 'appendCustomFile'
Output directory:
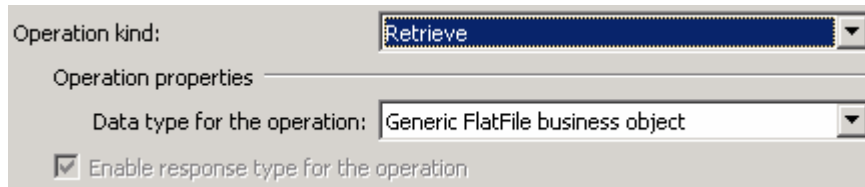Advanced >>

Add Operation: retrieveCustomFile

____ 16.   Click **Add…** to open Add Operation window

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

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

__ 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

Operation kind:	Retrieve	▼
Operation properties		
Data type for the operation:	Generic FlatFile business object	▼
☑ Enable response type for the operation		

__ 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 **retrieveCustomFile**

__ f. Define Data Binding type for **input**:

      1) Follow the instructions of defining data binding for createCustomFile and select **CustomDB** for input data format

__ g. Define Data Binding type for **output**:

      1) Accept the default Data format options selection, **Use data format configuration 'CustomDB'**

__ h. Define Output type:

      1) Under **Specify the operation output**, click **New…** next to **Output type** to open a New Business Object window

      2) Ensure that the Module selected is **FFCustomOutboundModule** and click **Next**

      3) From this window, Click **Browse…** next to **Data type**

      4) From the Data Type Selection window, select **Customer** under Matching data types and click **OK**

__ i. From the Business Object window, **check** the box next to **Generate business graph for each business object**

__ j. **Check** the box for **'Generate retrieve container to retrieve multiple business objects'**

Note: Selecting the **Generate retrieve container to retrieve multiple business objects** option creates **CustomerRetrieveWrapperBG**, which is used only if there are multiple business object records in the file that the adapter is going to retrieve. For this lab, you are going to retrieve a Customer business object that has two customer records.

__ k. Click **Finish.** You should see that the output type 'RetrieveResponseWrapper' is replaced with **CustomerRetrieveWrapperBG**
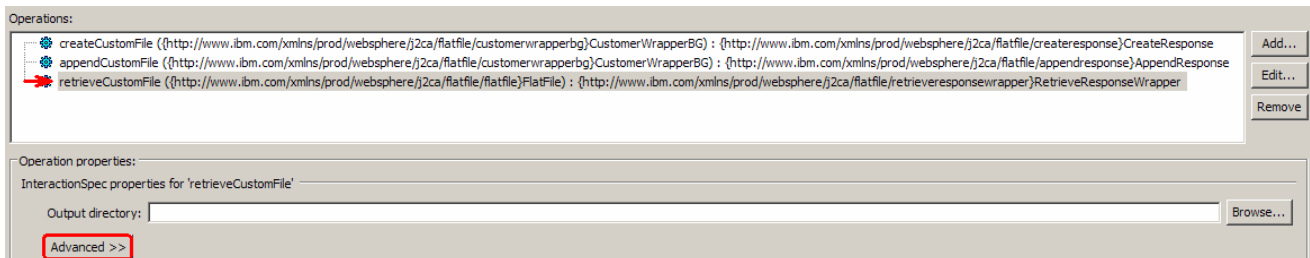
__ l. You should now see this:
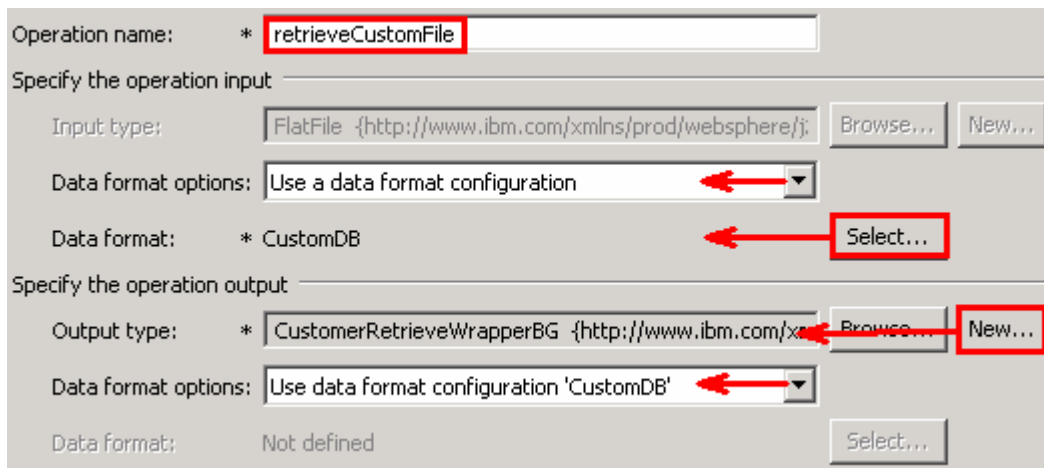
__ m. Click **Finish** from the Add Operation window

The operation, retrieveCustomFile, will now be displayed under Operations list.

____ 17. You can click **Advanced >>** under 'InteractionSpec properties for retrieveCustomFile' to review the properties available at Interaction spec level
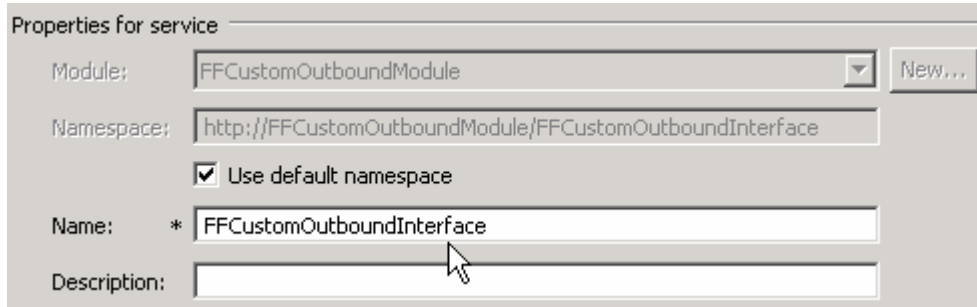

__ a. Click **Next** from the Operations window

____ 18. From the Generate Artifacts screen, enter these:

   __ a. For Name, enter **FFCustomOutboundInterface**

Properties for service		
Module:	FFCustomOutboundModule ▼	New...
Namespace:	http://FFCustomOutboundModule/FFCustomOutboundInterface	
	☑ Use default namespace	
Name: *	FFCustomOutboundInterface	
Description:		

   __ b. Click **Finish**

____ 19. You will now see a new import component, **FFCustomOutboundInterface** in the assembly diagram of FFCustomOutboundModule
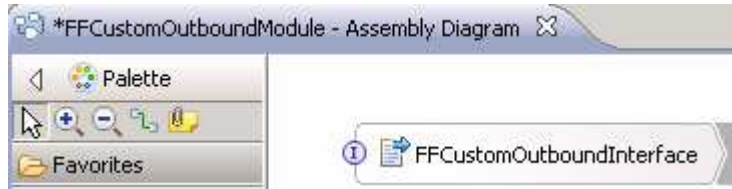
FFCustomOutboundModule - Assembly Diagram

◁ Palette
Favorites

ⓘ FFCustomOutboundInterface

   __ a. Save (**Ctrl+S)** your changes to the assembly diagram

____ 20. Review the FFCustomOutboundModule: The generated **Data Types**, **Interface**, and the Data binding (**CustomDB**), and Data handler (**CustomDH**) under Configured Resources can be found inside FFCustomOutboundModule
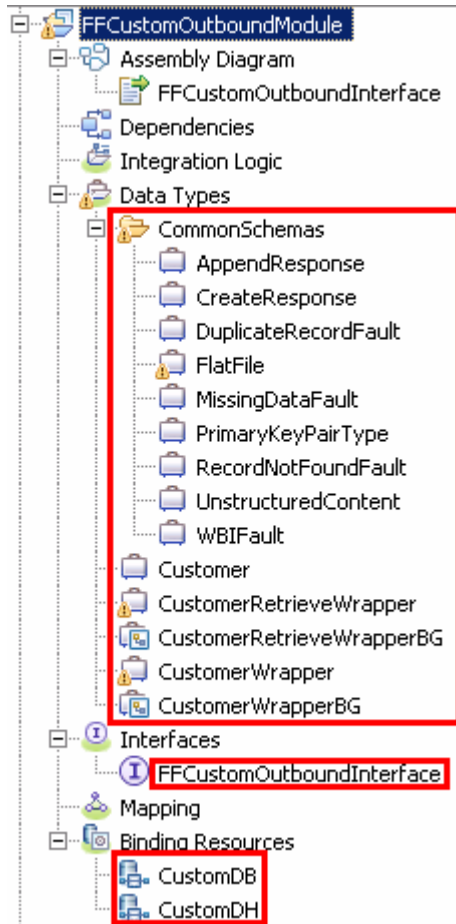
You can open each of these generated artifacts and business objects and review the properties inside.

Review the created methods inside the interface:

__ a. From the Business Integration view, expand FFCustomOutboundModule > Interfaces and then double-click **FFCustomOutboundInterface** to open it

__ b. You should see these three operations:

	Name	Type
▼ createCustomFile		
▷ Input(s)	createCustomFileInput	CustomerWrapperBG
▷ Output(s)	createCustomFileOutput	CreateResponse
✗ Fault	MISSING_DATA	MissingDataFault
✗ Fault	WBIFault	WBIFault
✗ Fault	DUPLICATE_RECORD	DuplicateRecordFault
✗ Fault	RECORD_NOT_FOUND	RecordNotFoundFault
▼ appendCustomFile		
▷ Input(s)	appendCustomFileInput	CustomerWrapperBG
▷ Output(s)	appendCustomFileOutput	AppendResponse
✗ Fault	MISSING_DATA	MissingDataFault
✗ Fault	WBIFault	WBIFault
✗ Fault	RECORD_NOT_FOUND	RecordNotFoundFault
▼ retrieveCustomFile		
▷ Input(s)	retrieveCustomFileInput	FlatFile
▷ Output(s)	retrieveCustomFileOutput	CustomerRetrieveWrapperBG
✗ Fault	MISSING_DATA	MissingDataFault
✗ Fault	WBIFault	WBIFault
✗ Fault	RECORD_NOT_FOUND	RecordNotFoundFault

__ c. Close the interface, FFCustomOutboundInterface

4.2.　　Test content specific 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 Process Server test environment

　__ a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects…** from the pop-up menu

　__ b. In the Add and Remove Projects window, select the **FFCustomOutboundModuleApp** project from the Available projects panel

　__ c. Click **Add >** to add it to the Configured projects panel

　__ d. The project is now moved to Configured projects. Click **Finish**

Wait for the project to be published to the server.

____ 3.　Open the test client for the module

　__ a. From the Business Integration perspective, right-click the **FFCustomOutboundModule** and select **Test > Test Module**

　__ b. The **FFCustomOutboundModule_Test** window is opened in the Assembly editor

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

- createCustomFile

- appendCustomFile

- retrieveCustomFile

Test Create operation:

____ 4.　Under **Detailed Properties**, for the **Operation** field, select **createCustomFile** from the drop down menu
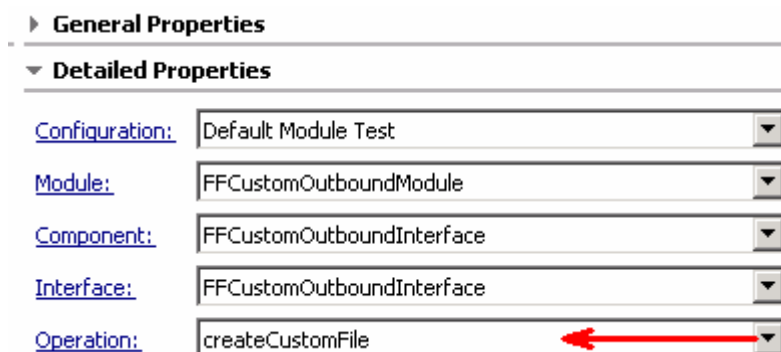
____ 5.   Fill out the fields for Initial request parameters:

__ a. For **fileName**, enter **UserDefined.xml**

__ b. For **includeEndBODelimiter**, enter **#####**

__ c. For **Content**, enter any random data. For Ex:

      1) Name: ABC

      2) Address: 11501 Burnet Rd

      3) City: Austin

      4) State: TX

Name	Type	Value
⊟ createCustomFileInput	CustomerWrapperBG	✔
verb	verb<string>	✔ CREATE
⊟ CustomerWrapper	CustomerWrapper	✔
directoryPath	string	✔
fileName	string	✔ UserDefined.xml
chunkFileName	string	✔
fileContentEncoding	string	✔
includeEndBODelimiter	string	✔ #####
stagingDirectory	string	✔
chunkNumber	string	✔
generateUniqueFile	boolean	✔ false
createFileIfNotExists	boolean	✔ false
splitFunctionClassName	string	✔
splitCriteria	string	✔
deleteOnRetrieve	boolean	✔ false
archiveDirectoryForDeleteOnRetrieve	string	✔
⊟ Content	Customer	✔
customerName	string	✔ ABC
Address	string	✔ 11501 Burnet Rd
City	string	✔ Austin
State	string	✔ TX

__ d. Click **Continue** button under Events

Events

    Invoke

WBPMv62_AdapterFFOutboundLab.doc

__ e. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server v6.2 at localhost** and click **Finish**
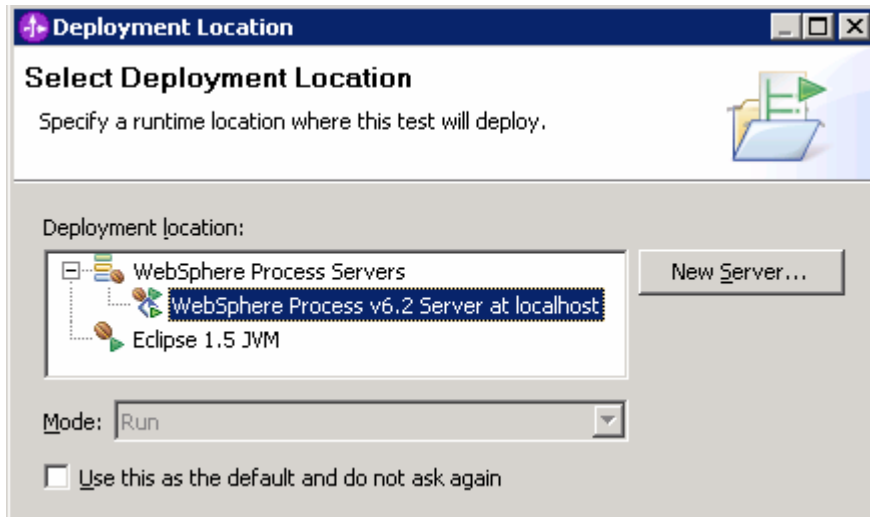
__ f. Provide Administrator User ID and Password

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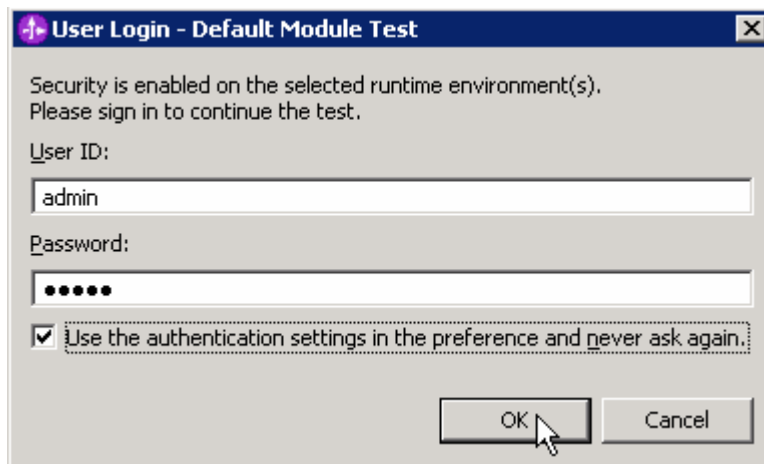
__ g. Click **OK** from Adding/Removing Projects pop-up, if asked. Optionally, you can select 'Do not show again' so that you are not required to do this next time when you start the test client

_____ 6. Verify the created file and its contents

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

__ b. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

__ c. You will see a new file, **UserDefined.xml** created under that directory. Double-click it to open the file. You should see the content that was entered and ##### at the end of the file.

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

Test Append operation:

_____ 7.   Click **Invoke** ( ) under Events to start a new event

_____ 8.   Under **Detailed Properties**, for the **Operation** field, select **appendCustomFile** from the drop down menu

Fill out the fields for Initial request parameters:

__ a. For **fileName**, enter **UserDefined.xml** (The file name should already exist for append operation. PassThroughTest.txt was created during Create operation test)

__ b. For **includeEndBODelimiter**, enter **#####**

__ c. For **Content**, enter any random data. For Ex:

> 1) Name: IBM

> 2) Address: 11901 RTP

> 3) City: Raleigh

> 4) State: NC

Name	Type	Value
verb	verb<string>	✔ CREATE
CustomerWrapper	CustomerWrapper	✔
directoryPath	string	✔
fileName	string	✔ UserDefined.xml
chunkFileName	string	✔
fileContentEncoding	string	✔
includeEndBODelimiter	string	✔ #####
stagingDirectory	string	✔
chunkNumber	string	✔
generateUniqueFile	boolean	✔ false
createFileIfNotExists	boolean	✔ false
splitFunctionClassName	string	✔
splitCriteria	string	✔
deleteOnRetrieve	boolean	✔ false
archiveDirectoryForDeleteOnRetrieve	string	✔
Content	Customer	✔
customerName	string	✔ IBM
Address	string	✔ 11901 RTP
City	string	✔ Raleigh
State	string	✔ NC

__ d. Click **Continue** button under Events

_____ 9.   Verify the results

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

    __ b. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

    __ c. You will see a new file, **UserDefined.xml** created under that directory. Double-click it to open the file. Observe the highlighted message appended to the original message.

Test Retrieve operation: In Retrieve operation, the adapter retrieves the file and parses it based on the configured SplittingFunctionClassName and SplitCriteria properties.

If splitting needs to be done based on file size, then splitFunctionClassName should be com.ibm.j2ca.utils.filesplit.SplitBySize which does the splitting functionality based on size and splitCriteria should be a number (a number representing the size in bytes). If the actual event file size is greater than this value, it is split into chunks else it is sent as one BO).

If splitting needs to be done based on a delimiter, then splitFunctionClassName should be com.ibm.j2ca.utils.filesplit.SplitByDelimiter which does this functionality and the splitCriteria (the delimiter which actually separates the BO's present in the event file) should be given.

This lab will guide you through the later splitting criteria, SplitByDelimiter.

____ 10. Click **Invoke** ( ) under Events to start a new event

____ 11. Under **Detailed Properties**, for the **Operation** field, select **retrieveCustomFile** from the drop down menu

Fill out the fields for Initial request parameters:

__ a. For **fileName**, enter **UserDefined.xml** (The file name should already exist for retrieve operation. UserDefinedTest.xml was created during Create operation test)

__ b. For **splitFunctionClassName**, enter **com.ibm.j2ca.utils.filesplit.SplitByDelimiter**

__ c. For **splitCriteria**, enter **#####**

Name	Type	Value
☐ 🔧 retrieveCustomFileInput	FlatFile	✔
⌐☐ directoryPath	string	✔
⌐☐ fileName	string	✔ UserDefined.xml
⌐☐ 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

__ d. Click **Continue** button under Events

____ 12. Verify the results:

__ a. You should see a window similar to this, which contains the content of the two business objects existing in the UserDefined.xml file:

Name	Type	Value
⊟ retrieveCustomFileOutput	CustomerRetrieveWrapperBG	✔
verb	verb<string>	✔
⊟ CustomerRetrieveWrapper	CustomerRetrieveWrapper	✔
⊟ [回] Content	Customer[]	6ᴨ
⊟ Content[0] ⬅	Customer	✔
customerName	string	✔ ABC
Address	string	✔ 11501 Burnet Rd
City	string	✔ Austin
State	string	✔ TX
⊟ Content[1] ⬅	Customer	✔
customerName	string	✔ IBM
Address	string	✔ 11901 RTP
City	string	✔ Raleigh
State	string	✔ NC

4.3. Restore server configuration

____ 1. Close the **FFCustomOutboundModule_Test** window and click **No** for the Save Resources window

____ 2. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects…** from the pop-up menu

____ 3. Select **FFCustomOutboundModuleApp** under Configured projects and click **< Remove**

____ 4. Click **Finish** after you see the application moved to Available projects. Wait until the application is unpublished

Part 5: Use default data binding

This part of the lab will show you how to use the default use the default function selector and data binding options from the External Service wizard and generate other required artifacts.

When you use the default function selector, you cannot define the rules as you did in Part 2 and hence there will only be one method that handles all types of files.

When you use the default data binding, you cannot have multiple data types as in Part 3 and each data type is handled by different method. Instead, there will only be one method and one data type.

After running the External Service wizard, you will continue to test the adapter.

5.1. Configure outbound using default function selector and data binding

In this part of the lab you will use the default function selector and data binding options from the External Service wizard and generate other required artifacts to test the outbound scenario.

____ 1.   Create FFDefaultsOutboundModule

___ a. From the Business Integration window, right-click and select **New > Module**

___ b. From the New Module window, enter **FFDefaultsOutboundModule** 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, FFDefaultsOutboundModule, created from your Business Integration window

____ 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)**
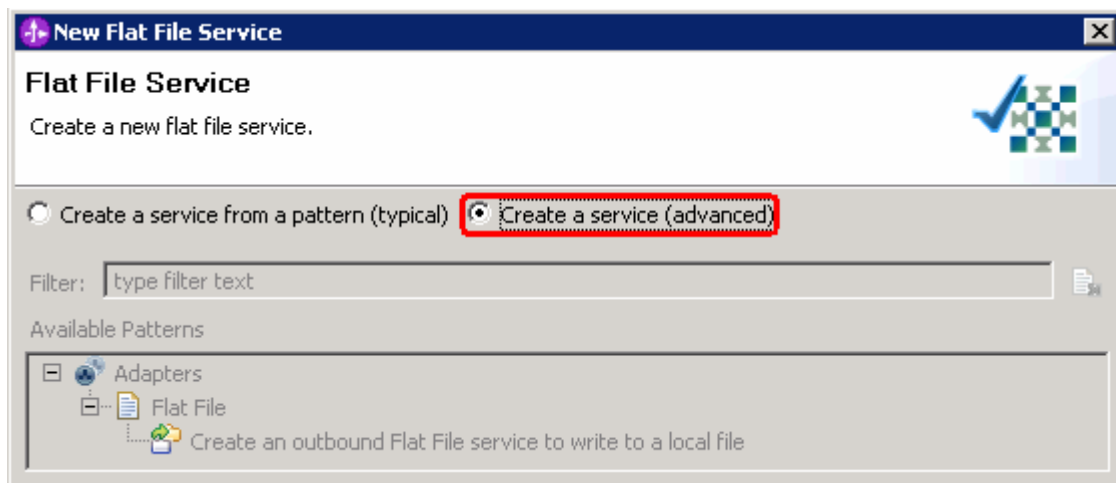
___ a. Click **Next**

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. The wizard provides three connectivity options – Adapters, Registers, and Messaging

Select the radio button next to **Adapters** 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**
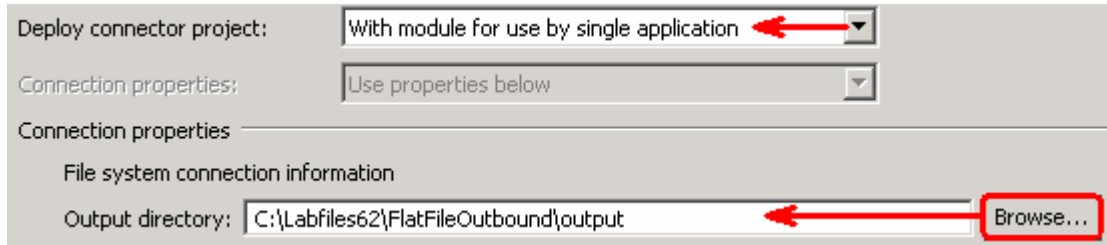
__ a. Click **Next**

____ 5.   Service Configuration Properties:

__ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected
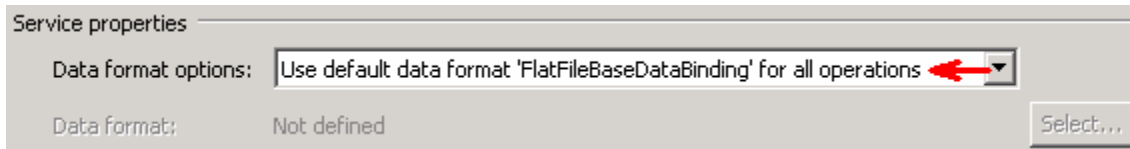
__ b. Click **Browse…** next to **Output Directory** and select **<OUTPUT_DIR>**

__ 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.

____ 6.   For **Data binding**, select **Use default data binding 'FlatFileBaseDataBinding' for all operations** 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**
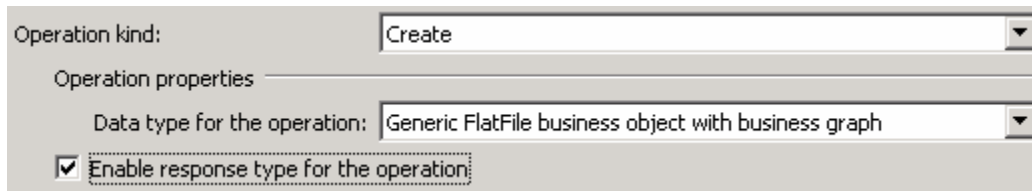
Define emitFlatFileBG operation:

____ 8.   From the Operations screen, click **Add…**

__ a. Add Operation window is opened. Select **Generic FlatFile business object with business graph** for the Data type

__ b. Check the box for '**Enable response type for the operation**'

__ c. Click **Next**

You are back to Operation window and because you have chosen the data type with business graph, the Input type is populated as **FlatFileBG**.

____ 9.   For **Operation name**, enter any name, for Ex: **createFlatFileBG**

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

____ 10. Accept the default selection, **Use suggested data binding 'FlatFileBaseDataBinding'**, as **Data format** for both Input and Output

__ a. Click **Finish**. The above defined operation, **createFlatFileBG**, is populated under Operations list

__ b. Click **Next** from Operations screen

____ 11. From Generate Service screen, accept the default value, **FlatFileImport**, for **Name**

__ a. Click **Finish**

____ 12. The Assembly diagram for FFDefaultsOutboundModule is opened with an Export component, **FlatFileImport**
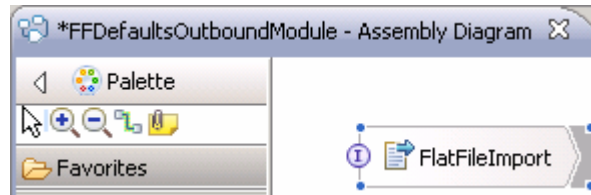
____ 13. Save (**Ctrl + S**) changes to your assembly diagram

5.2.　　Test defaults 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 pass through 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 **FFDefaultsOutboundModuleApp** under Available projects panel and click **Add >**

　__ c. You will now see the **FFDefaultsOutboundModuleApp** 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

____ 2.　Open the test client for the module

　__ a. From the Business Integration perspective, right-click the **FFDefaultsOutboundModule** and select **Test > Test Module**

　__ b. The **FFDefaultsOutboundModule_Test** window is opened in the Assembly editor

____ 3.　Under **Detailed Properties**, note that the Operation is **createFlatFileBG**

　__ a. Fill out the fields for Initial request parameters:

　__ b. For **fileName**, enter any name, for Ex: **DefaultsTest.txt**

__ c. For **AsText**, enter any string. For Ex: **Test Defaults scenario**

▸ **General Properties**

▾ **Detailed Properties**

Configuration:	Default Module Test ▾
Module:	FFDefaultsOutboundModule ▾
Component:	FlatFileImport ▾
Interface:	FlatFileImport ▾
Operation:	createFlatFileBG ▾

☐ Invoke export using binding

Initial request parameters

Name	Type	Value
⊟ createFlatFileBGInput	FlatFileBG	✔
└ verb	verb<string>	✔ CREATE
⊟ FlatFile	FlatFile	✔
├ directoryPath	string	✔
├ fileName	string	✔ DefaultsTest.txt
├ chunkFileName	string	✔
├ fileContentEncodir	string	✔
├ includeEndBODelin	string	✔
├ stagingDirectory	string	✔
├ chunkNumber	string	✔
├ generateUniqueFil	boolean	✔ false
├ createFileIfNotExi:	boolean	✔ false
├ splitFunctionClassI	string	✔
├ splitCriteria	string	✔
├ deleteOnRetrieve	boolean	✔ false
├ archiveDirectoryFc	string	✔
⊟ Content	UnstructuredContent	✔
├ ContentType	string	✔
├ ObjectName	string	✔
├ AsText	string	✔ Test Defaults scenario
└ AsBinary	hexBinary	✔ 0

__ d. Click **Continue** button under Events

__ e. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server v6.2** and click **Finish**

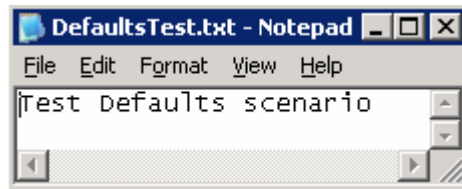__ f. You should see a window similar to this, which contains the data you just entered in the previous steps:
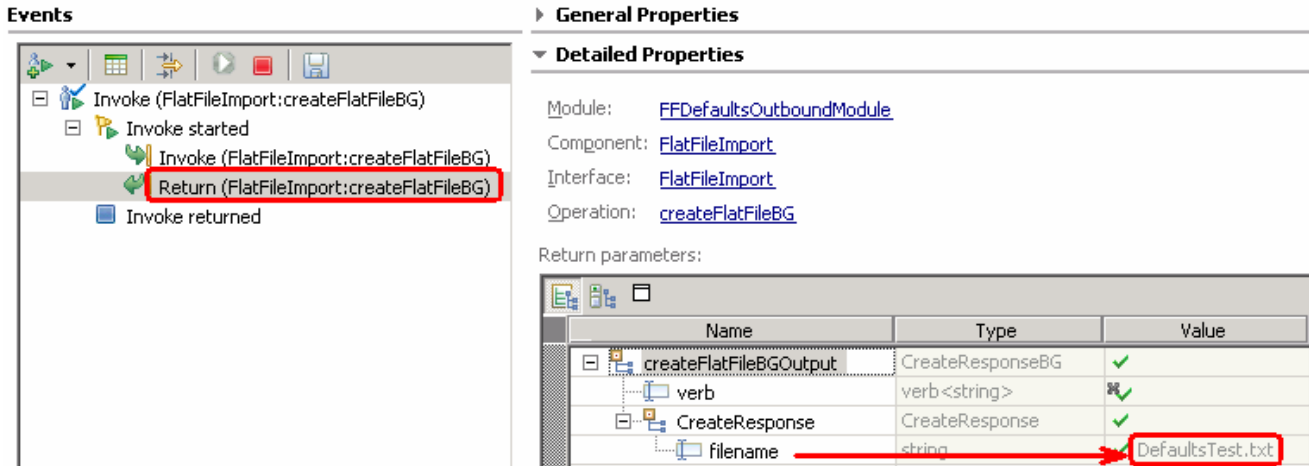
____ 4. To verify your test results, check the **<OUT_DIR>** subdirectory which should contain a file with the name **DefaultsTest.txt**.

____ 5. Double-click to open the file, DefaultsTest.txt and it should contain the text entered for AsText field in your test client

____ 6. Restore the Sever Configuration

__ a. Close the **FFDefaultsOutboundModule_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 **FFDefaultsOutboundModuleApp** 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

Part 6: Use 'Create a service from a typical pattern'

In this part of the lab you will use the **typical pattern** option from the External Service wizard to create and configure the Data Binding and other required artifacts to test the outbound scenario.

Based on your selection, the Binding resources (data binding) are created which you will review later in this part.

After running the External Service wizard, you will continue to test the adapter.

6.1.   Configure outbound using 'Create a service from a pattern (typical)' option

In this part of the lab you will use the **typical pattern** from the External Service feature to create and configure the Function Selector

____ 1.   Create the module: FFTypicalOutboundModule

__ a. From the Business Integration window, right-click and select **New > Module**

__ b. From the New Module window, enter **FFTypicalOutboundModule** 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, **FFTypicalOutboundModule**, created from your Business Integration window and the Assembly diagram for the same module is opened in the Assembly Editor.

____ 2.   Import required business objects

__ a. Expand FFTypicalOutboundModule (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 **FFTypicalOutboundModule** is selected for Into folder

__ f. Click **Finish** from the Import window

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

____ 3.   Review imported business object:

__ a. Expand **FFTypicalOutboundModule > Data Types** and you will now see a new data type **Customer** and **Order** under it.

__ b. Double-click **Customer** review the fields inside the object:
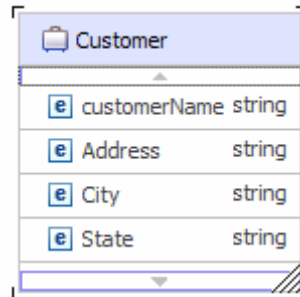
__ c. After reviewing, close the Customer business object from the Assembly editor

____ 4.   To start External Service from the Palette:

__ a. From the **Palette** on the left side of Assembly Diagram, click **Outbound Adapters**:

____ 5.   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

____ 6.   From the Flat File Service screen, accept the default selection of **Create a service from a pattern (typical)**
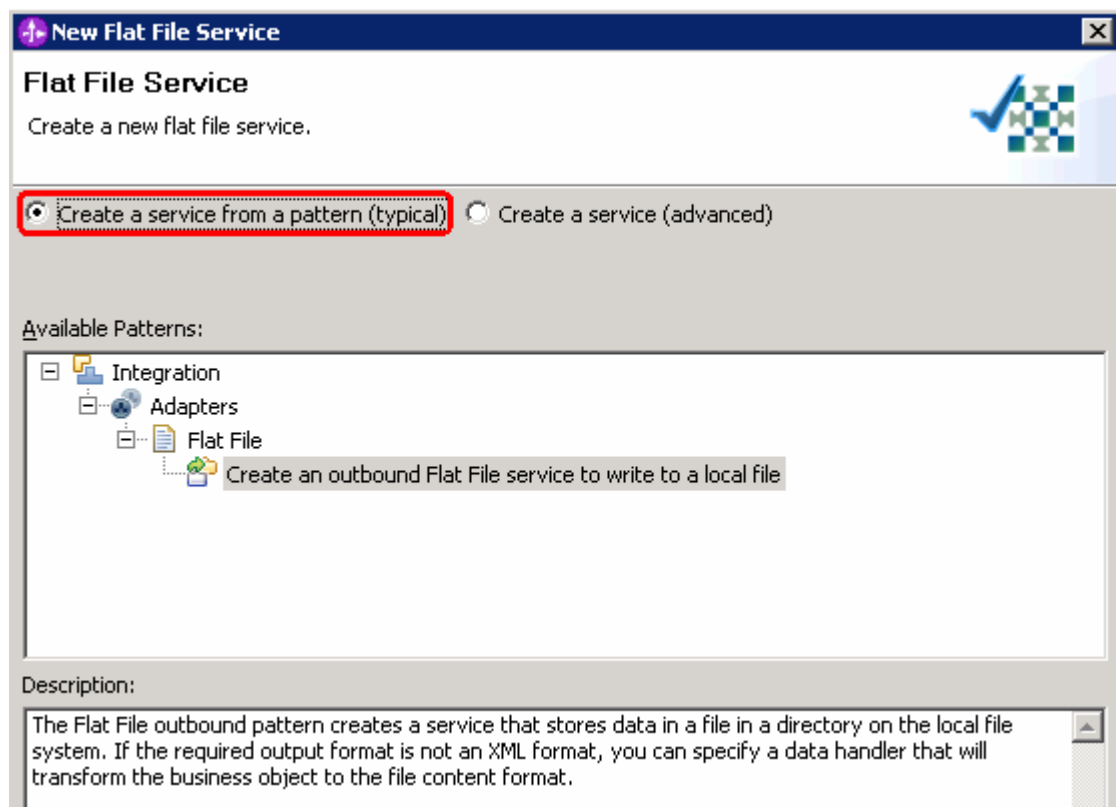
__ a. Click **Next**

IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

_____ 7. From the **Flat File service name** screen, for **Name**, accept the default value '**FlatFileImport**' and click **Next**

Module:	FFTypicalOutboundModule ▼ Browse... New...
Namespace:	http://FFTypicalOutboundModule/FlatFileImport ☑ Default
Folder:	Browse...
Name:	FlatFileImport

_____ 8. From the **Business object and directory screen**, enter these:

__ a. Click **Browse…** next to **Business object** and a Data Type Selection window is opened

__ b. Select **Customer** under Matching data types and click **OK**

**Data Type Selection**

Filter by type, namespace, or file (? = any character, * = any String):

```
*                                                                New...
```

Matching data types:

☐ Customer

__ c. Now, click **Browse…** next to **Output Directory** and a 'Browse For Folder' window is opened

__ d. From this window, navigate to select **<OUTPUT_DIR>** and click **OK**

__ e. Your Business object and directory screen should look like this:

New Outbound Flat File Service

Business object and output directory

Specify the business object and the output directory where the business object contents will be written.

What business object do you want to write to the output file?

Business object: Customer {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customer} ◄ Browse...

Where do you want to create the output file?

Output directory: C:\Labfiles62\FlatFileOutbound\output ◄ Browse...

__ f. Click **Next**

_____ 9. From the **Output file name** screen, enter these:

__ a. Select the radio button next to '**Generate a file name with an appended sequence number**'

__ b. Accept and note the default values for **File name**, **Directory**, and **Sequence file name**
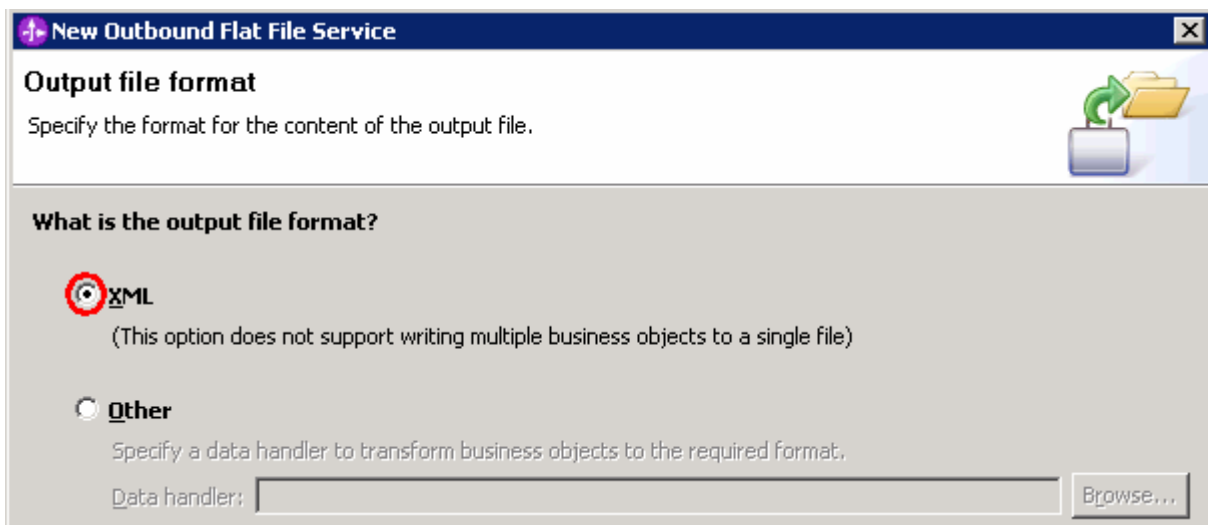

Note: If you select 'Generate a file name with and appended sequence number', the adapter will not generate the wrapper business object, instead uses the File name and Directory entered in this screen.
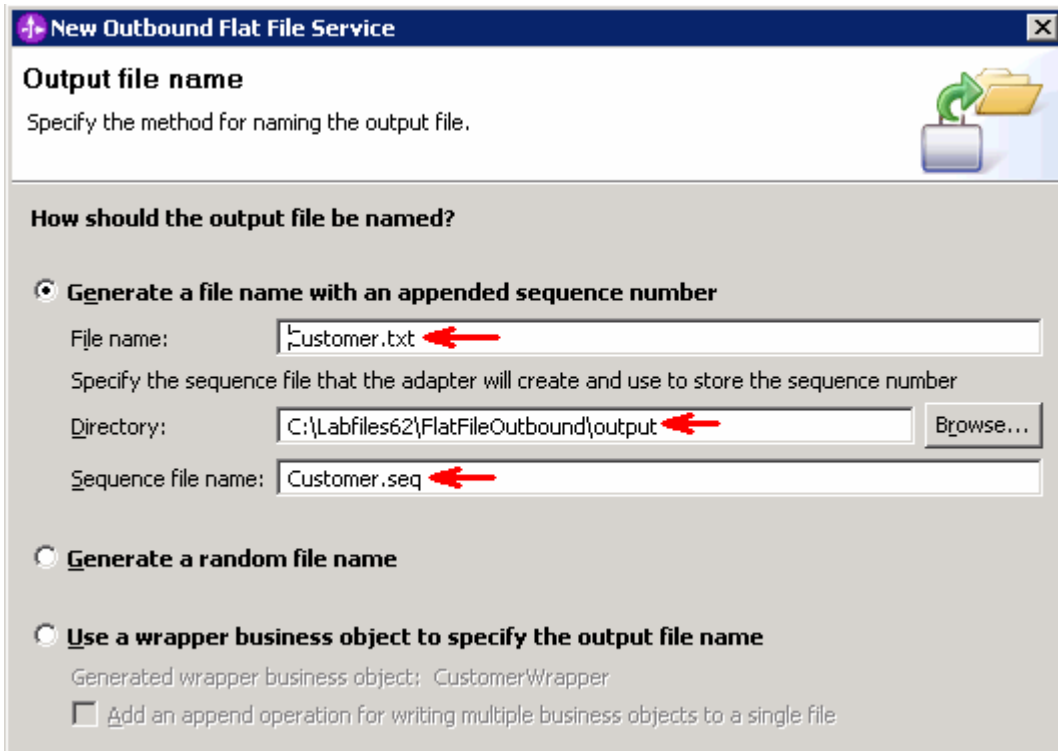
__ c. Click **Next**

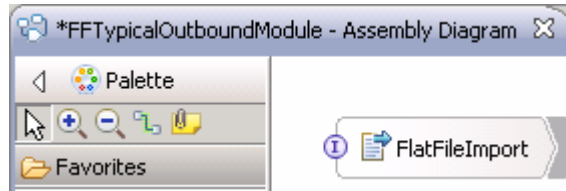____ 10. From the Output file format screen, enter these:

__ a. Select the radio button next to **XML**

__ b. Click **Finish**

WBPMv62_AdapterFFOutboundLab.doc

____ 11. Save (**Ctrl + S**) changes to your assembly diagram

____ 12. Review the FFTypicalOutboundModule and the generated artifacts: The generated **Data Types**, **Interface**, Data handler (**UTF8XMLDataHandler**) and Data binding (**FlatFileXMLDataBinding**) under Configured Resources can be found under FFTypicalOutboundModule. You can open each of these generated artifacts, business objects and review the properties inside.
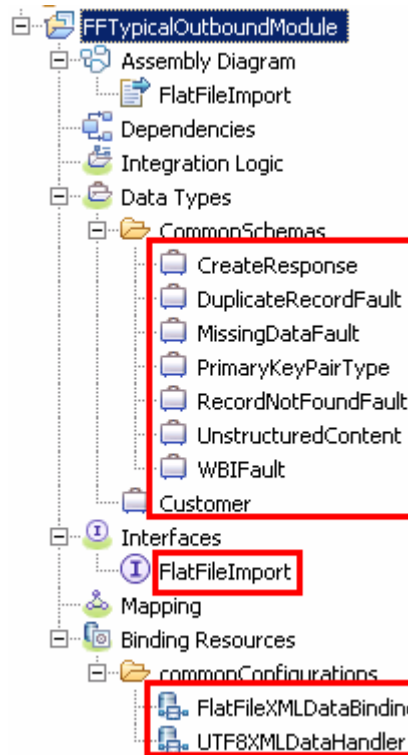
IBM WEBSPHERE ADAPTER 6.2 – LAB EXERCISE

6.2.   Test typical pattern 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 typical pattern with input file having single business object.

____ 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 **FFTypicalOutboundModuleApp** under Available projects panel and click **Add >**

    __ c. You will now see the **FFTypicalOutboundModuleApp** 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

Wait for the project to be published to the server.

____ 2.   Open the test client for the module

    __ a. From the Business Integration perspective, right-click the **FFTypicalOutboundModule** and select **Test > Test Module**

    __ b. The **FFTypicalOutboundModule_Test** window is opened in the Assembly editor

____ 3.   Under **Detailed Properties**, fill out these:

    __ a. Note that the **Operation** name is **create**

__ b. Enter these for Initial request parameters:
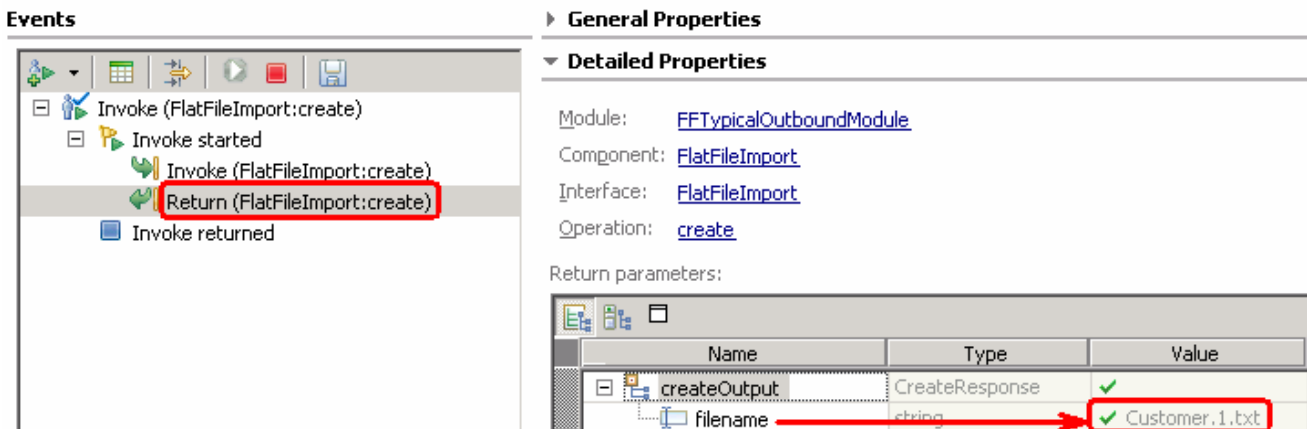
      1) customerName: ABC

      2) Address: 11501 Burnet Rd

      3) City: Austin

      4) State: TX
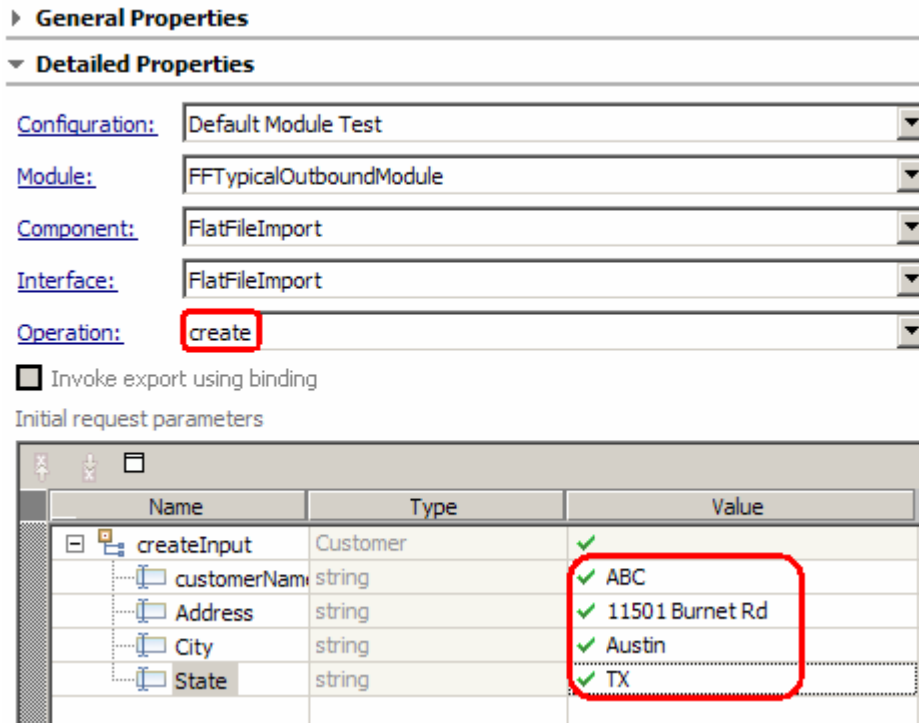
__ c. Click **Continue** button under Events

__ d. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server v6.2** and click **Finish**

__ e. You should see a window similar to this, which contains the data you just entered in the previous steps:
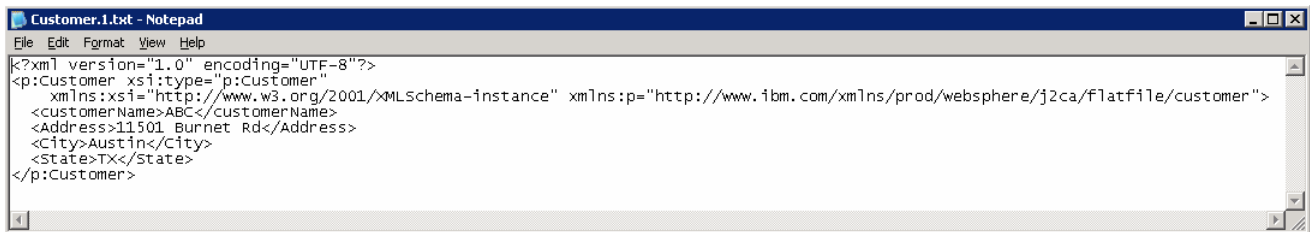
WBPMv62_AdapterFFOutboundLab.doc

Note: Since you have specified a Sequence File name while running the External Service, the file name is created with '1' appended to it.

_____ 4.   Verify the created file and its contents

    __ a. Open Windows Explorer and browse to the subdirectory **<OUT_DIR>**

    __ b. You will see a new file, **Customer.1.txt** created under that directory. Double-click it to open the file

```
Customer.1.txt - Notepad
File  Edit  Format  View  Help
<?xml version="1.0" encoding="UTF-8"?>
<p:Customer xsi:type="p:Customer"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customer">
  <customerName>ABC</customerName>
  <Address>11501 Burnet Rd</Address>
  <City>Austin</City>
  <State>TX</State>
</p:Customer>
```

_____ 5.   You should see one more file, **Customer.seq**, created in the same directory. Open it using note pad. You should see an entry in the file. For each request, the adapter increments the number in the sequence file and the input type takes the sequence number that is currently stored in the sequence file. Sequence numbers are not maintained separately for different input data types

_____ 6.   Restore the Sever Configuration

    __ a. Close the **FFTypicalOutboundModule_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 **FFTypicalOutboundModuleApp** 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

What you did in this exercise

In this lab, you imported the Flat File Adapter RAR file into your WebSphere Integration Developer workspace and integrated it into an SCA application that creates a file to the file system. 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 Operations which, after deploying onto the server, will generate Business Objects and other artifacts for different scenarios.

At the end of each part, you deployed and then tested the adapter application for the scenarios - pass-through test scenario (create, append, list, retrieve Operations), content specific or non pass through test scenario (create, append, and retrieve operations), using all defaults (default data binding) scenario (create operation), and then finally using the typical pattern (create operation).

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**.
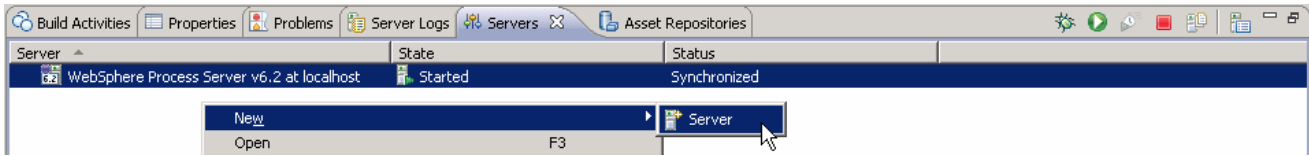

    __ c. In the New Server dialog, specify the remote server's host name, **<HOSTNAME>**.

    __ 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.**

WBPMv62_AdapterFFOutboundLab.doc
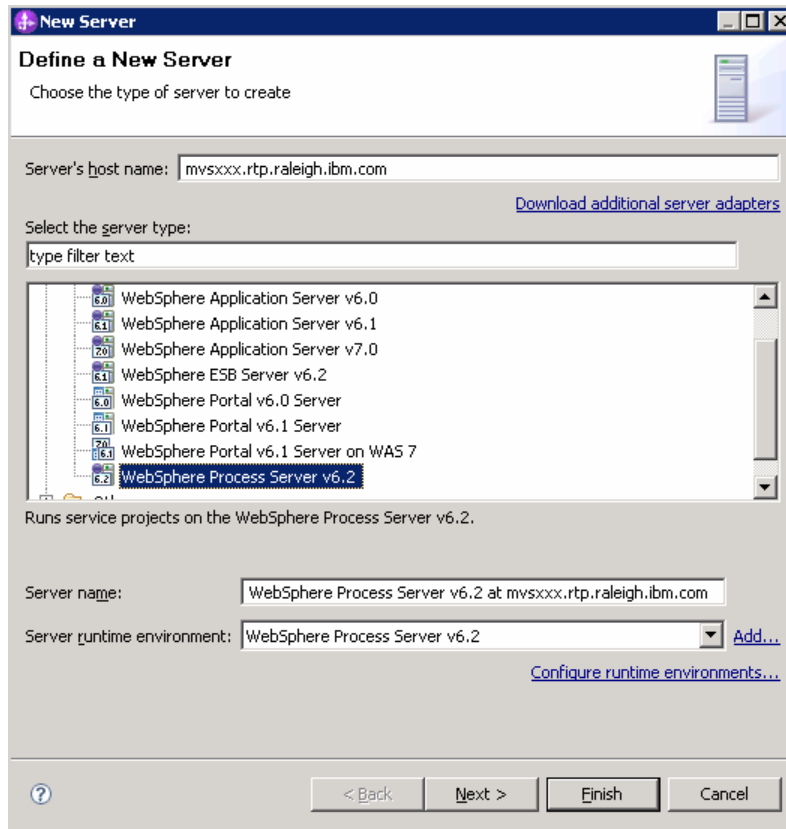
__ f. On the WebSphere Server Settings page, leave the radio button for **Manually provide connection settings** selected, and select the box for SOAP

__ 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.
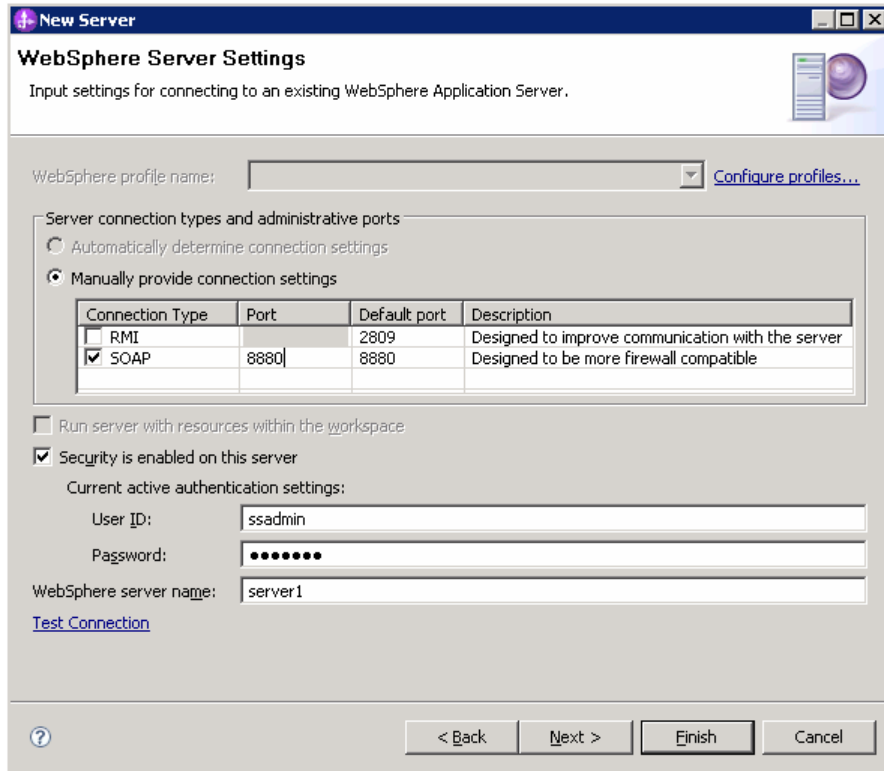
__ i. Click **Finish**.

__ 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 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
```