Flat File adapter – Processing COBOL copybook files lab

What th	is exercise is about1
Lab req	uirements1
What yo	bu should be able to do2
Introduc	2tion2
Exercis	e instructions2
Part 1: I	nitialize the workspace and prepare for the lab5
Part 2: I	nbound scenario
2.1.	Prepare for the inbound scenario7
2.2.	Configure inbound using the external service wizard12
2.3.	Add Java component17
2.4.	Test inbound scenario19
Part 3: 0	Outbound scenario21
3.1.	Prepare for the outbound scenario
3.2.	Configure outbound using external service wizard
3.3.	Test outbound scenario
Solution	n instructions
What yo	ou did in this exercise41
Task: A	dding remote server to WebSphere Integration Developer test environment

What this exercise is about

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

Lab requirements

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

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

What you should be able to do

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

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

Introduction

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

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

Exercise instructions

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

Reference variable	Windows location	AIX [®] or UNIX [®] location
<wid_home></wid_home>	C:\Program Files\IBM\WID7Beta	
<wps_home></wps_home>	C:\ <wid_home>\runtimes/bi_v70</wid_home>	
<ffadapter_home></ffadapter_home>	<wid_home>\ResourceAdapters\FlatFile_7.0.0.0\deploy</wid_home>	
<lab_files></lab_files>	C:\Labfiles70	/tmp/Labfiles70
<workspace></workspace>	<lab_files>\FlatFileCobol\workspace</lab_files>	
<event_dir></event_dir>	<lab_files>\FlatFileCobol\eventdir</lab_files>	
<archive_dir></archive_dir>	<lab_files>\FlatFileCobol\archivedir</lab_files>	
<out_dir></out_dir>	<lab_files>\FlatFileCobol\outdir</lab_files>	
<fffiles></fffiles>	<lab_files>\FFFiles</lab_files>	
<temp></temp>	C:\temp	/tmp

Windows users note: When directory locations are passed as parameters to a Java[™] program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\Labfiles70\ with C:/Labfiles70/

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></server_name>	server1	sssr011	
<was_home></was_home>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<hostname></hostname>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<soap_port></soap_port>	8880	8880	
<telnet_port></telnet_port>	N/A	1023	
<profile_name></profile_name>	AppSrv01	default	
<userid></userid>	N/A	ssadmin	
<password></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 extract the required lab files and start the WebSphere Integration Developer V7.0 with a new workspace and then define required WebSphere Environment Variables using the administrative console of WebSphere Process Server V7.0

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

🤀 Workspace Launcher	×
Select a workspace	
IBM WebSphere Integration Developer 7.0 stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
	Durau I
Workspace: C:\Labfiles70\FlatFileCobol\workspace	 Browse
▶ <u>C</u> opy Settings	
	Cancel

- ____ 3. Click the ^{_____} button on the right corner to close the Welcome page and proceed with the workbench
- ____4. Follow the instructions of "**Define WebSphere variable**" lab to define the WebSphere variables listed in the table below, which are used later in this lab:

Name	Value
FF_EVENT	C:\Labfiles70\FlatFileCobol\eventdir
FF_ARCHIVE	C:\Labfiles70\FlatFileCobol\archivedir
FF_OUT	C:\Labfiles70\FlatFileCobol\outdir

Part 2: Inbound scenario

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

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

2.1. Prepare for the inbound scenario

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

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

🤀 New Module 📃 🗖 🗙
Create a Module Use a module to integrate business applications and services. A module can contain any type of component and can be deployed on WebSphere Process Server.
Module name: FFCOBOLInboundModule
Location: C:\Labfiles70\FlatFileCobol\workspace\FFCOBOLInboundModule Browse

- ____d. Click Finish
- ____ e. You will now see a new module, FFCOBOLInboundModule, created in your Business Integration window
- 2. Generate the business object definition for the COBOL program source file
 - ____a. In the Business Integration section of the window, right-click the module, FFCOBOLInboundModule, and select **New > Business Object from External Data**

____b. From the Business Object from External Data window, select Languages > COBOL

🚯 Business Object From External Data	
Input Sources For Creating Business Objects	
Select an input source for creating business objects.	
Filter: type filter text	
Input Sources:	
E Adapters	
C ⊂	
C (CICS Channel)	
C (Multiple Output)	
COBOL	
COBOL (CICS Channel)	-
Description:	
Create a new business object by importing data structures from a COBOL file.	A

___ c. Click Next

____d. In the Business Object Mapping Details window,

1) Note that the Selected mapping value is COBOL to Business Object

2) For COBOL file, click Browse... and select the traderc99FF.ccp file located in <FFFILES>

🥵 External Data Discovery	
Business Object Mapping Details The selected mapping details for the business objects you want to create.	0,
Selected mapping: COBOL to Business Object	V
COBOL file:* C:\Labfiles70\FFFiles\taderc99FF.ccp <	Browse

3) Click Next

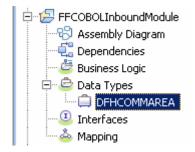
____e. In the Select Data Structures window, the new business object, called **DFHCOMMAREA**, is displayed

🚯 External Data	×
Select Data Structures	
Specify the settings used to compile the COBOL, C or PL/I file. Then click Find to discover the data structures in the file and select from which to build business objects.	\bigcirc
Platform: Win32	
Code page: ISO-8859-1	Select
Advanced >>	
Data structures:	
(1) DFHCOMMAREA	Find

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

🚯 External Data		×
Generate Busi Specify the name ar	ness Objects Id location of the new business object.	H
Module: Namespace:	FFCOBOLInboundModule http://FFCOBOLInboundModule Use default namespace	New
Folder: Name: * Generation Style:	DFHCOMMAREA	Browse
0	< <u>B</u> ack <u>N</u> ext > Einish	Cancel

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



- ____3. Generate a wrapper business object definition. Wrapper business object definitions wrap existing business object definitions with additional function.
 - ____a. In the Business Integration section of the window, right-click the module and select New > Business Object from External Data
 - ____b. From the External Data window, select Adapters > Flat File under Available Types:

🤂 Business Object From External Data 📃	Ι×
Input Sources For Creating Business Objects Imput Source for Creating Business Objects Select an input source for creating business objects. Imput Source for Creating Business Objects)
Eilter: type filter text	$\mathbf{B}_{\mathbf{R}}$
Input Sources:	
Adapters Adapters Flat File FIP FTP	
E-E Languages	•
Description:	
Create a new business object using the FlatFile resource adapter to wrap an existing business object and dynamically specify interaction values.	4

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

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

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

2) Select DFHCOMMAREA under Matching data types

🚯 Data Type Selection 📃 🗖 🗙
Filter by type, namespace, or file (? = any character, $*$ = any String):
*
Matching data types:

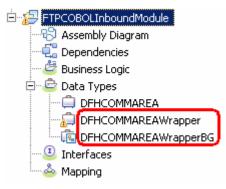
3) Click OK

- ____e. To generate a business graph, select the Generate business graph for each business object check box
- _____f. Do not select the Generate retrieve container to retrieve multiple business objects check box. This new options is used and explained in outbound scenario

🚯 New Business Object	
Business Object Properties	
Specify the properties for the new business object.	
Data type: * DFHCOMMAREA {http://FFCOBOLInboundModule}	Browse New
Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile	
Califerate retrieve container to retrieve inditiple business objects.	

___ g. Click Finish

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



2.2. Configure inbound using the external service wizard

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

- ____1. Ensure that the FFCOBOLInboundModule assembly diagram is opened in Assembly editor
 - ____a. In the Business Integration view, expand FFCOBOLInboundModule and double-click Assembly Diagram
- 2. To start External Service from the Palette:
 - ____a. From the **Palette** on the left side of Assembly Diagram, click **Inbound Adapters**:
 - 3. Under Inbound Adapters, click the **Flat File** and then click the empty canvas of the assembly diagram



4. From the New External Service window, expand Adapters > Flat File and select Simple: Create an inbound Flat File service to read from a local file

🚯 New External Service 📃 🗖 🗙
Select the Service Type or Registry 🛛 🙀 🥢
Select the type of service to create or registry to browse.
Eilter: type filter text
<u>A</u> vailable types:
🖂 💣 Adapters
🗄 📄 Flat File
Advanced: Create a Flat File service using the complete wizard
Simple: Create an inbound Flat File service to read from a local file
Description:
The Flat File inbound pattern creates a service that retrieves a file from a directory on the local file system. If the file is not in an XML format, you can specify a data handler that will transform from the file content format to business objects. The file content can be split if the content contains multiple copies of the data structure for processing.

_ a. Click Next

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

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

Select Adapters > Flat File and click Next

_ 5. On the Select an Adapter screen, select IBM WebSphere Adapter for Flat Files (IBM : 7.0.0.0) > CWYFF_FlatFile and click Next



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

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

Deploy connector projec	t: > [With module for use by single application	•	
Connection properties:	Ī	Use properties below	~	
Connection properties -				
File system connect	ion information:			
Event directory:*	\${FF_EVENT}			Browse

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

<pre>(<< Advanced)</pre>
Event polling configuration
Event delivery configuration
Event persistence configuration
Advanced Properties
File archiving configuration
Bidi properties
Logging and tracing

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

- ____d. Event polling configuration: This has all the polling configuration details and for this lab, you can accept the defaults.
- ____e. Event delivery configuration:
 - Ensure once-only event delivery: You should check this box only if you are using data source and table name in the Event persistence configuration (below). If this property is set to true, while using in-memory capability (explained below), the adapter will log a warning message. By default this is selected and you can accept the default selection.
- _____f. Event persistence configuration: In this part of the lab, you will not use any JNDI instead use adapter's in-memory representation of event table to store all the necessary information

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

- ____g. Advanced properties:
 - 1) Poll subdirectories in event directory:

Note: **'Poll subdirectories in event directory'** is the new Boolean property added in V6.2. This property is used for recursive polling. Flat File adapter polls the given 'Event directory' and all the sub-directories present in the 'Event directory'. Refer to **'Log and confidential trace lab'** for more details on this new feature

- ___h. File archiving configuration:
 - 1) Archive directory: **\${FF_ARCHIVE}**

 File archiving config 	juration		
Specify archive dire	ectory to archive	processed inbound event files	
Archive directory:		\${FF_ARCHIVE}	Browse
File extension for a	archive:	original]
Success file extens	ion for archive:	success]
Failure file extension	on for archive:	fail]

- 7. Under Service properties, for Function selector options, select Use default function selector 'FilenameFunctionSelector' from the drop down list
- 8. For Data format options, select Use COBOL, C or PL/I data format from the drop down list

5er	vice properties		
	Function selector options:	Use default function selector 'FilenameFunctionSelector'	
	Function selector:	Not defined	Select
	Data format options:	Use COBOL, C or PL/I data format	
	Data format:	Not defined	Select

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

Define emitCOBOLFile operation:

- _____ 10. From the Operations screen, click Add...
 - ____a. Add Operation window is opened. Select **User Defined Type** for the Data type and click **Next**

Operation properties	
The data type for the operation input:	User defined type

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

____12. For Input type, click Browse... and select DFHCOMMAREAWrapperBG from the Data Type Selection window and click OK

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

Operation name:*	emitCOBOLFile		
Specify the operation	on input		
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com/xme	wse	New

____a. Click Finish. The defined operation, emitCOBOLFile, is populated under Operations list

Operatio	ons:	
····· 🔅	emitCOBOLFile ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperbg}DFHCOMMAREAWrapperBG)	(Add)

____b. Click **Next** from Operations screen

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

Properties for ser-	vice -	
Module:	FFCOBOLInboundModule	New
Namespace;	http://FFCOBOLInboundModule/FlatFileExport	
	☑ Use default namespace	
Name: *	FlatFileExport	
Description:		

___a. Click Finish

_____14. The Assembly diagram for FFCOBOLInboundModule is opened with an Export component, FlatFileExport:

🕄 *FFCOBOLInboundMo	dule - Assembly Diagram 🛛
👌 😲 Palette	
💫 🗨 🗨 🖫 🕼	
🕞 Favorites	FlatFileExport

____a. Save (Ctrl + S) changes to your assembly diagram

2.3. Add Java component

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

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

👌 😴 Palette
\$®(€(°⊾ <mark>©</mark> ,
🔁 Favorites
🕞 Components
💁 Untyped Component
👔 Human Task
🚰 Java
Mediacon Flow
👤 Proce Java
E Rule Group
State Machine

- ____b. Click **Java** and then click the empty space of FFCOBOLInboundModule assembly diagram. This will place a new component, **Component1** on the assembly diagram.
- 3. Wire the FlatFileExport to the Component1
 - ___a. Select the wire (1) icon from the Palette
 - ____b. Click FlatFileExport and then click Component1 to wire them together
 - ____ c. Select **OK** for the Add Wire pop-up window:

🚯 Add Wire 🛛 🛛 🔀
This action will allow the target service to be used in other modules. The service interface from the export will be added to the target. Do you want to continue?
Always create without prompt
Cancel

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

Your assembly diagram for FFCOBOLInboundModule will look like this:

🕄 *FFCOBOLInboundModule - Assembly Diagram 🛛					
👌 😳 Palette					
🔓 🗨 🗨 🖫 🕼					
🗁 Favorites	🚺 📸 FlatFileExport				

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

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

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

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

2.4. Test inbound scenario

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

- 1. Add the project to the WebSphere Test Environment server
 - ____a. Right-click **WebSphere Process Server v7.0** under the Servers view and select **Add and remove projects...** from the pop-up menu
 - ____ b. From the Add and Remove Projects window, select FFCOBOLInboundModuleApp under Available projects panel and click Add >
 - ____ c. You will now see the FFCOBOLInboundModuleApp added to the Configured projects
 - _____d. Click **Finish** and wait until the project is being published onto the server. The server will start in Debug mode if it is not already started before
 - 2. Test the application by providing input files:

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

- ____a. Copy the **COBOLTest.txt** file from <FFFILES> to <EVENT_DIR>. The adapter will poll the copied file from the event directory and will transfer it to the archive directory
- ____b. It will pass through the **emitFlatFileBG** method and you should see this message in your **Server** Logs view (or SystemOut.log):

🔀 Task Flows 🕜 Build Activities 🔲 Properties 🔝 Problems 👔 Server Logs 🛛 🖓 Servers 🖪 Asset Repositories						
				🗄 🖻 🕆 🕂 🚭		
WebSphere Process Server v7.0 at localhost 🕴						
Console (filtered): V	VebSphere Process Serv	erv7 flati	ocalhost			
oonooic (intereu). +	repopliere i locess dell	CI #1.0 dt1	Jean of the second s			
Show All Record Types (Hierarchical) > with only Server State and Error Contents (Page 1 of 1)						
Туре	▲ Time	Thread ID	Contents			
🔲 🗐 Log message	Dec 1, 2009 17:40:12.765	0000004f	WSVR0200I: Starting application: AppScheduler			
🔲 🗐 Log message	Dec 1, 2009 17:40:13.828	0000004f	WSVR02211: Application started: AppScheduler			
🔲 🗐 Log message	Dec 1, 2009 17:40:14.468	00000051	WSVR0221I: Application started: BusinessRules_qnode_serv			
🔲 🗐 Log message	Dec 1, 2009 17:40:14.765	0000004f	WSVR0200I: Starting application: ivtApp			
🔲 🗐 Log message	Dec 1, 2009 17:40:14.921	00000051	WSVR02001: Starting application: ESBSamplesGallery			
🔲 💷 Log message	Dec 1, 2009 17:40:16.218	00000051	WSVR02211: Application started: ESBSamplesGallery			
🔲 📑 Log message	Dec 1, 2009 17:40:16.640	0000004f	WSVR02211: Application started: ivtApp			
🔲 🗐 Log message	Dec 1, 2009 17:40:17.484	00000050	WSVR02211: Application started: BPMAdministrationWidgets			
🔲 🗐 Log message	Dec 1, 2009 17:40:18.078	00000051	WSVR02001: Starting application: TaskContainer_gnode_ser			
🔲 🗐 Log message	Dec 1, 2009 17:40:29.859	00000051	WSVR02211: Application started: TaskContainer_qnode_ser			
🔲 🗐 Log message	Dec 1, 2009 17:40:32.156	00000000	WSVR00011: Server server1 open for e-business			
🔲 🗐 Log message	Dec 3, 2009 19:36:29.656	0000009d	WSVR02001: Starting application: IBMUTC			
🔲 💷 Log message	Dec 3, 2009 19:36:34.078	0000009d	WSVR02211: Application started: IBMUTC			
🔲 📑 Log message	Dec 3, 2009 19:37:22.359	000000bd	WSVR02001: Starting application: FFCOBOLInboundModuleApp			
🔲 🗐 Log message	Dec 3, 2009 19:37:37.890	000000bd	WSVR02211: Application started: FFCOBOLInboundModuleApp			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	**************************************			
🔲 🖃 Log message	Dec 3, 2009 19:37:56.734	00000084	File Name : COBOLTest.txt			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	NAME> ABC7			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	First Name> James			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	Last Name> Bond			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	Street> Burnet Rd			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	City> Austin			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	Country> USA			
🔲 🗐 Log message	Dec 3, 2009 19:37:56.734	00000084	Phone> 5128380000			
🗖 📑 Log message	Dec 3, 2009 19:37:56.734	00000084	Postal Code> 78758			

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

Address 🛅 C:\Labfiles70\FlatFileCobol\archivedir					
Folders	×	Name 🔺	Size	Туре	
E 🛅 FlatFileCobol		COBOLTest.txt.2009_12_03_19_37_56_046.success	1 KB	SUCCESS File	

_____d. Open the file using note pad or word pad and verify the contents of it

COBOLTest.txt.2009_12_03_19_37_56_046.success - WordPad						
<u>File E</u> dit <u>V</u> iew <u>I</u> nsert	F <u>o</u> rmat <u>H</u> elp					
ABC7 James	Bond	Burnet Rd	Austin	USA	5128380000	78758

_____ 3. Restore the Sever Configuration

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

Part 3: Outbound scenario

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

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

3.1. Prepare for the outbound scenario

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

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

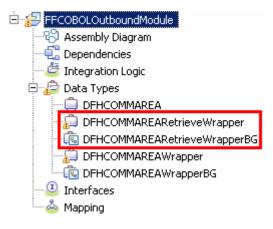
You will now see a new module, FFCOBOLOutboundModule, created from your Business Integration window

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

🚯 New Business Object	
Business Object Properties Specify the properties for the new business object.	€
Data type: * DFHCOMMAREA {http://FFCOBOLOutboundModule} ✓ 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/flatf ✓ Generate retrieve container to retrieve multiple business objects.	Browse New

___ g. Click Finish

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



3.2. Configure outbound using external service wizard

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

- 1. Ensure that the FFCOBOLOutboundModule assembly diagram is opened in Assembly editor
 - ____a. In the Business Integration view, expand FFCOBOLOutboundModule and double-click Assembly Diagram
- 2. To start External Service from the Palette:
 - ____a. From the Palette on the left side of Assembly Diagram, click Outbound Adapters
 - ____b. Under Outbound Adapters, click the **Flat File** and then click the empty canvas of the assembly diagram. The New Flat File Service wizard is opened
 - _ 3. From the New External Service window, expand Adapters > Flat File and select Advanced: Create a Flat File service using the complete wizard

🥵 New External Service 📃 🗖 🗙
Select the Service Type or Registry Select the type of service to create or registry to browse.
Eilter: type filter text
Available types:
 Adapters Flat File Advanced: Create a Flat File service using the complete wizard Simple: Create an outbound Flat File service to write to a local file
Description:
Creates a service that reads from or writes to a file on a local file system using the WebSphere® Adapter for Flat Files. You can create integrated processes, which include the exchange of data with the local file system, without special coding. During outbound processing, services running on WebSphere Process Server or WebSphere Enterprise Service Bus use the adapter to perform operations on files in the local file system, while during inbound processing, these services use the adapter to receive events from the local file system.
a. Click Next

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

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

Select Adapters > Flat File and click Next.

- ____4. On the Select an Adapter screen, expand IBM WebSphere Adapter for Flat Files (IBM : 7.0.0.0) 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. For Output Directory enter \${FF_OUT}
 - ____ c. Click **Advanced >>** to see the hidden advanced properties that can be configured:

<< Advanced			
Logging and tracing			
Advanced properties			
Bidi properties			

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

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

Service properties					
Data format options:	Use COBOL, C or PL/I data format				
Data format:	Not defined	Select			

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

Define createCOBOL operation:

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

🚯 Add Operation		
Operation Specify the properties for the operation	on to add.	5
Operation kind:	Create	
Operation properties		
	User defined type	←
🔽 Enable response type for the	operation	

____ d. Click Next

- 9. For **Operation name**, enter any name, for Ex: **createCOBOL**
- 10. For Input type, click **Browse** and select **DFHCOMMAREAWrapperBG** from the Data Type Selection window

Operation name: *	createCOBOL				
Specify the operati	on input				
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com/xnterBrowse New				
Specify the operation output					
Output type:	CreateResponse {http://www.ibm.com/xmlns/prod/we Browse New				

____a. Click Finish. The defined operation, createCOBOL, is populated under Operations list

1	Operations:	
	🏶 createCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperb	Add

Define appendCOBOL operation:

- _____ 11. From the Operations screen, click Add...
 - ____a. In the Add Operation window, select **Append** from the drop down menu for Operation kind
 - ____b. Select **User Defined Type** for the Data type
 - _____c. Select the check box next to Enable response type for the operation
 - ____d. Click Next
 - ____e. For **Operation name**, enter **appendCOBOL**
 - ____ f. For Input type, click Browse and select DFHCOMMAREAWrapperBG from the Data Type Selection window

Operation name: *	appendCOBOL		
Specify the operat	ion input		
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com	Browse	New
Specify the operat	ion output		
Output type;	AppendResponse {http://www.ibm.com/xmlns/prod/we	Browse	New

____g. Click Finish. The defined operation, appendCOBOL, is populated under Operations list

Operations:	
createCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperb	Add
appendCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperl	Edit
	Remove
Operation properties:	
InteractionSpec properties for 'appendCOBOL'	
Output directory: Bro	owse
Advanced >>	

Define retrieveCOBOL operation:

- _____ 12. Click **Add...** to open Add Operation window
 - ____a. For **Operation kind**, select **Retrieve** from the drop down list
 - ____b. For Data type for operation, select Generic FlatFile business object from the drop down list
 - ____ c. Note that the box next to Enable response type for the operation is checked by default

Operation kind:	Retrieve
Operation properties	
Data type for the operation:	Generic FlatFile business object
$\overline{\mathbf{V}}$ 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 retrieveCOBOL
- ____f. Define Output type:
 - 1) Under **Specify the operation output**, click **Browse...** next to **Output type** to open a Data Type Selection window

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

🚯 Data Type Selection 📃 🗖 🗙
Filter by type, namespace, or file (? = any character, $*$ = any String):
*
Matching data types:
DFHCOMMAREA
DFHCOMMAREARetrieveWrapper
DFHCOMMAREAWrapper

3) Click OK

____ g. You should now see this:

🔂 Add Operation	
Operation	
Specify the properties for the operation to add.	
Operation name: * retrieveCOBOL	
Specify the operation input	
Input type: FlatFile {http://www.ibm.com/xmlns/prod/websphere/j2ca/l	Browse, New,
Specify the operation output	
Output type:* DFHCOMMAREARetrieveWrapperBG {http://www.ibmem/-	Browse New

- ___h. Click **Finish** from the Add Operation window
- _____i. You can click **Advanced >>** under 'InteractionSpec properties for retrieveCOBOL' to review the properties available at Interaction spec level

Operations:	
reateCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperb	Add
appendCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/dfhcommareawrapperl	
retrieveCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/flatfile}FlatFile) : {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/flatfile}FlatFile) : {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile}FlatFile}FlatFile) : {http://www.ibm.com/xmlns/prod/websphere/j2ca/flatFile}FlatFile}FlatFile}FlatFile	Edit
	Remove
٠ () () () () () () () () () (
Operation properties:	
InteractionSpec properties for 'retrieveCOBOL'	
Output directory: Bro	wse
Advanced >>	

- ----

- ____j. Click **Next** from Operations screen
- _____ 13. From Generate Service screen, accept the default value, FlatFileImport, for Name

Properties for se	rvice		
Module:	FFCOBOLOutboundModule	-	New
Namespace;	http://FFCOBOLOutboundModule/FlatFileImport		
	🔽 Use default namespace		
Name:	* FlatFileImport		
Description:			

___a. Click Finish

_____14. The Assembly diagram for FFCOBOLOutboundModule is opened with an Export component, FlatFileImport

*FFCOBOLOutboundM	odule - Assembly Diagram 🛛 🕅
👌 👯 Palette	
ନୁ ି୍ ୍ ୍ %	1 mar
🔁 Favorites	FlatFileImport

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

3.3. Test outbound scenario

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

- 1. Start WebSphere Process Server (if not started already)
 - ____a. From the **Servers** view of WebSphere Integration Developer, right click **WebSphere Process** Server v7.0 and select Start from the pop-up menu
 - ____b. Wait until the server status shows as Started
- _____2. Add the project to the WebSphere test environment server
 - ____a. Right-click **WebSphere Process Server v7.0** under the Servers view and select **Add and remove projects...** from the pop-up menu
 - ____ b. From the Add and Remove Projects window, select FFCOBOLOutboundModuleApp under Available projects panel and click Add >
 - ____ c. You will now see the FFCOBOLOutboundModuleApp added to the Configured projects
 - _____d. Click **Finish** and wait until the project is being published onto the server. The server is started in Debug mode if it is not already started before
- _____ 3. Open the test client for the module
 - ____a. From the Business Integration perspective, right-click the **FFCOBOLOutboundModule** and select **Test > Test Component**
 - ____b. The FFCOBOLOutboundModule_Test window is opened in the Assembly editor

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

- createCOBOL
- appendCOBOL
- retrieveCOBOL

Test Create operation:

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

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. <u>More...</u>

Configuration:	Default Module Test	•
Module:	FFCOBOLOutboundModule	•
Component:	FlatFileImport	•
Interface:	FlatFileImport	•
Operation:	createCOBOL	•

Initial request parameters:

⊙ Value editor O XML editor

Ę	± E		8
	Name	Туре	Value
	🖃 🏪 createCOBOLInput	DFHCOMMAREAWrapperBG	ab 🔺
	🖳 💭 verb	verb <string></string>	I CREATE
	🗄 🖳 DFHCOMMAREAWrapper	DFHCOMMAREAWrapper	в
	🖳 💭 directoryPath	string	ав
	🖳 💭 fileName	string	📧 COBOLTest.txt
	🖳 🚛 chunkFileName	string	ав
	🖳 💭 fileContentEncoding	string	ав
	includeEndBODelimite	string	▣ ###

_____d. Enter the values of your choice for the fields under Content

🗄 📲 Content	DFHCOMMAREA	аы
🖳 💭 CustomerNumber *	CustomerNumber <string></string>	B ABC7
💭 FirstName *	FirstName <string></string>	🗈 James
💭 LastName *	LastName <string></string>	📧 Bond
	Street <string></string>	📧 Burnet Rd
抑 City *	City <string></string>	🗈 Austin
💭 Country *	Country <string></string>	III USA
···· 💭 Phone *	Phone <string></string>	5128380000
	PostalCode <string></string>	78758

____e. Click **Continue** button under Events

🛉 🕨 Invoke

Events	
	iys the events in a test trace. Select an event to display its e General Properties and Detailed Properties sections. <u>More</u>
	▶ - = - =

_____f. From Deployment Location window, select WebSphere Process Servers > WebSphere Process Server v7.0 at localhost and click Finish

🔒 Deployment Location	
Select a Deployment Location Specify a runtime location where this test will deploy.	E
Deployment location:	
🖃 🖶 WebSphere Process Servers	New <u>S</u> erver
WebSphere Process Server v7.0 at localhost	
Eclipse 1.5 JVM	
WebSphere Enterprise Service Bus Server	
Mode: Run	
🔲 Use this location as the default and do not ask again	

- ____g. Provide Administrator User ID and Password
 - 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

🌐 User Login - Default Module Test 🛛 🛛 🔀
Security is enabled on the selected runtime environment(s). Sign in to continue the test.
<u>U</u> ser ID;*
admin
Password:*
•••••
Use the authentication settings in the preference and never ask again
OK Cancel

___h. Verify the results:

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

Events	General Properties					
This area displays the events in a test trace. Select an	 Detailed Properties 					
event to display its properties in the General Properties and Detailed Properties sections. <u>More</u>	<u>M</u> odule:	FFCOBOLOutboundModule	<u>e</u>			
▶ ■ ≱ • = • ⊇ 券 🖫	Component:	<u>FlatFileImport</u>				
Invoke (FlatFileImport:createCOBOL)	Interface:	<u>FlatFileImport</u>				
🖃 💦 Invoke started	Operation:	<u>createCOBOL</u>				
🔄 Invoke (FlatFileImport:createCOBOL)	Return parameters:					
♀< Return (FlatFileImport:createCOBOL)	Value Editor	XML Source				
Invoke returned		NINE DOGICO				
	Er Br	•				
		Name	Туре	Value		
		createCOBOLOutput	CreateResponse	(36)		
		💭 filename	string	📧 COBOLTest.txt		

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

Address 🛅 C:\Labfiles70\FlatFileCobol\outdir						
Folders		x	Name 🔺	Size	Туре	
	FlatFileCobol		E COBOLTest.txt	1 KB	Text Document	

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

📕 COBOLTest.txt - I	Notepad			
<u>File E</u> dit F <u>o</u> rmat <u>y</u>	/iew <u>H</u> elp			
ABC7 James 5128380000	Bond 78758 ###	Burnet Rd	Austin	USA 🔺

Test Append operation:

- _____ 5. Add invoke
 - ____a. Click the invoke button under Events to add a new invoke

Events		
event to dis	play it	the events in a test trace. Select an s properties in the General Properties and s sections. <u>More</u>
	å⊳ ₽	- ■ - □ ‡ □

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

General Properties

Detailed Properties

Specify the component, interface, operation, and input parameter values for the Invoke event, then click the Continue icon in the Events area to run the test. <u>More...</u>

Configuration:	Default Module Test	•
Module:	FFCOBOLOutboundModule	•
Component:	FlatFileImport	•
Interface:	FlatFileImport	•
Operation:	appendCOBOL	•

Initial request parameters:

0	Value editor O XML editor			
Ę	🔹 🛛 🗖			
	Name	Туре	Value	
	🖻 💾 appendCOBOLInput	DFHCOMMAREAWra	аь	
	🖳 💭 verb	verb <string></string>	CREATE	
	🗄 🖳 DFHCOMMAREAWrap	DFHCOMMAREAWra	аы	
	🖳 🛄 directoryPath	string	ар	
	🖳 💭 fileName	string	📧 COBOLTest.txt	
	🖳 🛄 chunkFileName	string	35	
	🖳 💭 fileContentEncodi	string	аы	
		string	⊡b ###	ļ

____ d. Enter the values of your choice for the fields under Content

🗄 🖳 🔁 Content	DFHCOMMAREA	ab
💭 CustomerNumber *	CustomerNumber <string></string>	📧 XYZ9
師 FirstName *	FirstName <string></string>	🗈 James
💭 LastName *	LastName <string></string>	🗈 Doe
🏧 Street *	Street <string></string>	📧 Duvan Rd
🛄 City *	City <string></string>	🗈 Austin
💭 Country *	Country <string></string>	III USA
抑 Phone *	Phone <string></string>	5128380000
🛄 PostalCode *	PostalCode <string></string>	78727

____e. Click Continue button under Events

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

Events	General Properties				
area displays the events in a test trace. Select an event to display its					
properties in the General Properties and Detailed Properties sections. More		<u>M</u> odule:	FECOBOLOutboundMod	<u>ule</u>	
		Compone	nt: FlatFileImport		
Invoke (FlatFileImport:createCOBOL) Rev Rev Started		Interface	ElatFileImport		
Invoke started Invoke (FlatFileImport:createCOBOL)		Operation	appendCOBOL		
Return (FlatFileImport:createCOBOL)	Return parameters:				
Invoke returned	Value Editor XML Source				
 □ M Invoke (FlatFileImport:appendCOBOL) □ N Invoke started 		E: 8:			
Invoke (FlatFileImport:appendCOBOL)			Name	Туре	Value
Return (FlatFileImport:appendCOBOL)			appendCOBOLOutput	AppendResponse	ав
Invoke returned	1		🛄 filename	string	📧 COBOLTest.tx

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

📕 COBOLTest.txt ·	- Notepad						_ 🗆 🗡
<u>File E</u> dit F <u>o</u> rmat	<u>V</u> iew <u>H</u> elp						
ABC7 James	Bond		Burnet	Rd	Austin	USA	
5128380000	78758 ###XYZ9		Doe		Duval Rd		
Austin	USA	5128380000	78727	###			

Test Retrieve operation:

- 7. Click the invoke button under Events to add another invoke
- 8. Under **Detailed Properties**, note that the Operation is **retrieveCOBOL**
 - ____a. Fill out the fields for Initial request parameters:
 - ____b. For filename, enter COBOLTest.txt
 - ____c. splitFunctionClassName: com.ibm.j2ca.utils.filesplit.SplitByDelimiter
 - ____d. splitCriteria: ###

Name	Туре	Value
🖃 🖳 retrieveCOBOLInput	FlatFile	ав
🖳 İirectoryPath	string	ав
🖳 fileName	string	📧 COBOLTest.txt
🖳 🛄 chunkFileName	string	ав
🏣 fileContentEncoding	string	аь
💭 includeEndBODelimiter	string	ав
🏣 stagingDirectory	string	ав
💭 chunkNumber	string	ав
🖳 💭 generateUniqueFile	boolean	🕮 false
抑 createFileIfNotExists	boolean	🕮 false
抑 splitFunctionClassName	string	💷 com.ibm.j2ca.utils.filesplit.SplitByDelimiter
💭 splitCriteria	string	
🖳 💭 deleteOnRetrieve	boolean	🕮 false
- I archiveDirectoryForDeleteOnRetrieve	string	ав
🗄 🖳 Content	UnstructuredContent	ав
ContentType	string	ав
	string	ав
AsText	string	ав
🛄 AsBinary	hexBinary	IDD 00

____e. Click Continue button under Events

____ f. Verify the results:

1) There is a new return under retrieveCOBOL invoke

Events

This area displays the events in a test trace. Select an event to display its properties in the General Properties and Detailed Properties sections. <u>More...</u>

→ ■ 🏞 • 🎞 • 🖻 券 🔛
🖃 🐩 Invoke (FlatFileImport:createCOBOL)
🖃 隆 Invoke started
🔄 Invoke (FlatFileImport:createCOBOL)
✓I Return (FlatFileImport:createCOBOL)
🔲 Invoke returned
🖂 🐩 Invoke (FlatFileImport:appendCOBOL)
🖂 隆 Invoke started
💚 Invoke (FlatFileImport:appendCOBOL)
✓I Return (FlatFileImport:appendCOBOL)
🔲 Invoke returned
🖃 👬 Invoke (FlatFileImport:retrieveCOBOL)
🖃 隆 Invoke started
🐂 Invoke (FlatFileImport:retrieveCOBOL)
Keturn (FlatFileImport:retrieveCOBOL)
Invoke returned

2) Select the Return and notice the return parameters, which contains the data you just entered in the previous steps:

Return parameters

Name	Туре	Value	
🖃 🖳 retrieveCOBOLOutput	DFHCOMMAREARetrieveWrapperBG	ав	
師 verb	verb <string></string>	86	
🗄 📲 DFHCOMMAREARetrieveWrapper *	DFHCOMMAREARetrieveWrapper	аь	
É…[□] Content	DFHCOMMAREA[]	ഒറ	
🚊 📲 Content[0] 🔫 — — —	DFHCOMMAREA	аы	
CustomerNumber *	CustomerNumber <string></string>	ABC7	
💭 FirstName *	FirstName <string></string>	🗈 James	
抑 LastName *	LastName <string></string>	Bond	
抑 Street *	Street <string></string>	💷 Burnet Rd	
江 City *	City <string></string>	ab Austin	
抑 Country *	Country <string></string>	LED USA	
💭 Phone *	Phone <string></string>	□□ 5128380000	
PostalCode *	PostalCode <string></string>	ab 78758	
🗄 📲 Content[1] 🔫	DFHCOMMAREA	аы	
CustomerNumber *	CustomerNumber <string></string>	III XYZ9	
💭 FirstName *	FirstName <string></string>	💷 James	
抑 LastName *	LastName <string></string>	Doe	
抑 Street *	Street <string></string>	🗈 Duvan Rd	
(ity *	City <string></string>	ab Austin	
Country *	Country <string></string>	LED USA	
抑 Phone *	Phone <string></string>	■ 5128380000	
	PostalCode <string></string>	ab 78727	

- 9. Restore the Sever Configuration
 - ____a. Close the **FFCOBOLOutboundModule_Test** window and click **No** for the Save Resources window
 - ____b. Right-click **WebSphere Process Server v7.0** under the Servers view and select **Add and remove projects...** from the pop-up menu
 - ____ c. Select **FFCOBOLOutboundModuleApp** under Configured projects and click **< Remove**
 - ____d. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished

Solution instructions

- _____1. Start WebSphere Integration Developer V7.0 with a new workspace
 - ____a. Follow the instructions outlined in Part 1 of this exercise
- _____ 2. Import the Project Interchange
 - ____a. Import the project interchange file FFCOBOL_PI.zip from <LAB_FILES>\FlatFileCobol\solution directory
 - ____b. Select File → Import from the menu
 - ____ c. Select Other → Project Interchange in the Import dialog and click Next

🌐 Import	_ 🗆 🗙
Select Import a project and its dependent projects from a Zip file.	ù
Select an import source: type filter text	_
 Test Web Web services Mu Mu Other FTP HTTP Localized Model Project Interchange 	
⑦ < Back Next > Einish	Cancel

- ____d. For the From zip file, click on the Browse button and select the FFCOBOL_PI.zip in the <LAB_FILES>>\ FlatFileCobol\solution directory
- ____e. Enter <LAB_FILES>\FlatFileCobol\workspace for the Project location root

____f. Click the **Select All** button. This will select all the modules: **CWYFF_FlatFile**, **FFCOBOLInboundModule**, and **FFCOBOLOutboundModule**

🚯 Import Project In	terchange Contents	
Import Projects Import Projects from a	a zip file.	<u> </u>
From zip file: Project location root:	C:\Labfiles70\FlatFileCobol\solution\FFCOBOL_PI.zip C:\Labfiles70\FlatFileCobol\workspace	Browse
Germann Germannn Germannnn Germannn Germannn Germannn Germannnn Germannnnn Germannnnn Germannnn	bundModule	
Select All Deselec	t All Select Referenced	
0	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

- ___ g. Click Finish
- _____ 3. Continue with **Part 2.4** of this lab to test the inbound scenario
- 4. Continue with **Part 3.3** of this lab to test the outbound scenario

What you did in this exercise

In this lab, you started with importing the Flat File Adapter RAR file into your WebSphere Integration Developer new workspace. Next, you made use of the External Data wizard to generate business object and wrapper definitions from a COBOL program source file. Then you continued to with External Service wizard available in WebSphere Integration Developer to generate other artifacts for inbound and then for outbound.

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

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.

🔀 Task Flows 🗟 Build Activities 🔲 Properties 🗌	🐮 Problems 👔 Server Logs 👫 Serve	rs 🛛 🚺 Asset Repositories 🛛 🗖 🗖
		参 🜔 🍠 📒 🔁 🛅
Server A	State	Status
🔣 WebSphere Process Server v7.0 at localhos	·····	
	Ne <u>w</u>	🕨 👕 Server 🔪
	Open	F3

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

🚯 New Server
Define a New Server 🔤 🧧
Choose the type of server to create
Server's host name: mvsxxx.rtp.raleigh.ibm.com
Download additional server adapters
Select the server type:
type filter text
WebSphere Business Monitor Server v7.0 on WebSphere Application Server WebSphere Business Monitor Server v7.0 on WebSphere ESB WebSphere Business Monitor Server v7.0 on WebSphere Process Server WebSphere ESB Server v7.0 WebSphere Portal v6.0 Server WebSphere Portal v6.1 Server WebSphere Portal v6.1 Server on WAS 7 WebSphere Process Server v7.0 WebSphere Process Server v7.0
Runs service projects on the WebSphere Process Server v7.0.
Server name: WebSphere Process Server v7.0 at mvsxxx.rtp.raleigh.ibm.com
Server runtime environment: WebSphere Process Server v7.0
Configure runtime environments

___e. Click Next

- _____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.

🌐 New Server				
WebSphere Application Server Settings Input settings for connecting to an existing WebSphere Application Server.				
Profile name:	Profije name: Configure profiles			
Server connection types a C Automatically determin Manually provide conn	ne connection s	settings		
Connection Type	Port	Default port	Description	
IPC		9633	Recommended for local servers	
	8880	2809	Designed to improve communication wit Designed to be more firewall compatible	
	0000	0000	Designed to be more nirewaii compatible	
Run server with resource	es within the <u>v</u>	vorkspace		
🔽 Security is enabled on t	his server			
Current active authent	ication setting:	s:		
User <u>I</u> D:	ssadmin			
Pa <u>s</u> sword:				
Application server name:	server1			
Test Connection				

____ i. Click Finish.

____j. The new server should be seen in the Server view.

🔀 Task Flows 🗟 Build Activities 🔲 Properties 🔝 Problems 🔞 Se	rver Logs 🤼 Servers 🛛	🔓 Asset Repositories 🛛 🗖 🗖
		🌣 🜔 🖉 🔲 🔡 🛅
Server 🔺	State	Status
🔀 WebSphere Process Server v7.0 at localhost	🖡 Started	Synchronized
WebSphere Process Server v7.0 at mvsxxx.rtp.raleigh.ibm.com	🛃 Stopped	Synchronized

- 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>

____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

This page is left intentionally blank.