# FTP adapter – Processing COBOL copybook files lab

# 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 FTP 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 FTP 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 <sup>®</sup> or UNIX <sup>®</sup> location
<wid_home></wid_home>	C:\Program Files\IBM\WID70	
<wps_home></wps_home>	C:\ <wid_home>\runtimes/bi_v70</wid_home>	
<ftpadapter_home></ftpadapter_home>	<wid_home>\ResourceAdapters\FTP_7.0.0.0\deploy</wid_home>	
<lab_files></lab_files>	C:\Labfiles70	/tmp/Labfiles70
<workspace></workspace>	<lab_files>\FTPCobol\workspace</lab_files>	
<event_dir></event_dir>	<lab_files>\FTPCobol\eventdir</lab_files>	
<archive_dir></archive_dir>	<lab_files>\FTPCobol\archivedir</lab_files>	
<ftpfiles></ftpfiles>	<lab_files>\FTPFiles</lab_files>	
<temp></temp>	C:\temp	/tmp

**Windows users' note**: When directory locations are passed as parameters to a Java<sup>™</sup> 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 table below is related to where you are running the remote test environment:

Reference variable	Example: Remote Windows test server location	Example: Remote z/OS <sup>®</sup> 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 V6.2 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. Create directory structure on your FTP Server
  - \_\_\_\_a. Log onto FTP machine/FTP Server using your ftp user and its password
  - \_\_\_\_b. Create these directories under the user's home directory:
    - 1) mkdir EventDir
    - 2) mkdir ArchiveDir
    - 3) mkdir **OutDir**
  - \_\_\_\_3. 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.	•.	
Workspace: C:\Labfiles70\FTPCobol\workspace	-	Browse
▶ <u>Copy Settings</u>		
?	Ж	Cancel

\_\_\_\_ 4. Click the <sup>\_\_\_\_</sup> button on the right corner to close the Welcome page and proceed with the workbench

\_\_5. 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:

**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.

Name	Value
FTP_EVENT	/root/EventDir
FTP_ARCHIVE	/root/ArchiveDir
FTP_LOCAL_EVENT	C:\Labfiles70\FTPCobol\LocalEventDir
FTP_LOCAL_ARCHIVE	C:\Labfiles70\FTPCobol\LocalArchiveDir
FTP_OUT	/root/OutDir

# 1.1. Review new properties

This part of the lab will give you brief description of some the new properties that are not used in this lab.

- New in V7.0: FTPS Connection mode: Specify the connection mode (Implicit or Explicit) to define the data protection level with which the data is transferred between the client and the server can be configured.
  - Explicit In an explicit mode, the connection begins with an unencrypted FTP connection. When any sensitive information such as password needs to be sent, the client explicitly issues a request to switch to a secure FTP connection. After the successful SSL negotiation, a secure command channel is established between the client and the server. Explicit mode works with the default port 21 and is compliant with RFC 2228 commands. RFC 2228 specifies the mechanism for authenticating connections and confidential data transfer between the client and the server and this is referred as explicit mode. The AUTH command is used for specifying the security mechanism for the explicit mode. The client sends AUTH command (AUTH SSL/TLS) to the FTPS server and switches to a secure command connection.
  - Implicit In an implicit mode, the communication between the client and server is setup immediately in secure mode. The text information exchanged between the client and server is in an encrypted form. The default port for implicit mode is 990.
- New in V7.0: Data channel protection level:
  - **Private** Data is transferred in encrypted form: Select Private, if the data transfer between the Adapter and the FTPS server should be in an encrypted form
  - Clear Data is transferred as clear text: Select Clear, if the data transfer between the Adapter and the FTPS server should be in clear text form

Note: The default value is set to Private

**Note: Server authentication** can be enabled while establishing a SSL connection. While using SSL, FTPS server sends its certificate to the FTP client for verification. The FTP client verifies the certificate to ascertain that it is communicating with the intended FTP server. To enable this verification process, the FTP server's certificate should be present in the client's trust store. Following three properties are used for this server authentication.

- New in V7.0: Keystore type: Specify type of the keystore
  - o JKS
  - o PKCS12

**Note**: Use keytool utility, if you want to import server's certificate into client's trust store. For example, enter the command keytool -import -v -alias serverCert -file server.cert -keystore clientTrustStore where server.cert is the certificate of the server and clientTrustStore is the trust store of the client. Ensure that the value of Keystore type property is same as the type used while creating the keystore.

- New in V7.0: Truststore file\*: Specify the path of the truststore file that contains the certificates of the servers trusted by the adapter
- New in V7.0: Truststore password: Specify the password of the truststore. It is used to check the
  integrity of the truststore data. If the value is not specified, the integrity check is not performed. It is
  applicable only if the protocol value is set to FTP over SSL or FTP over TLS

**Note: Client authentication** can be enabled while establishing a SSL connection. When using SSL/TLS, FTPS server requests for the client's certificate. The FTPS server verifies the certificate sent by the client to ascertain that it is communicating with the intended client. To enable this verification process, the FTPS server has to support client authentication and the client's certificate should be present at the server's trust store. At the client's end, client's keystore information has to be available for the exchange of the certificate to take place. Following three properties are used for this client authentication along with the Keystore type.

- New in V7.0: Keystore file: Specify the path of the keystore file. The keystore file will contain the
  private key entry of the FTPS client. It is also accompanied by a certificate chain for the corresponding
  public key
- New in V7.0: Keystore password: Specifiy the password of the keystore. It is used to check the
  integrity of the keystore data. If the value is not specified, integrity check is not executed. It is
  applicable only if the protocol value is set to FTP over SSL or FTP over TLS
- New in V7.0: Key password: Set the Key password to the password provided while creating the key in the keystore. This value is required to extract the certificate from the keystore while establishing a SSL connection.

Note: All these properties are applicable only if the protocol value is set to FTP over SSL or FTP over TLS

• New In V7.0: Enable remote verification: It checks whether the control and data connections are established with the same host (typically the machine from which you establish a connection to the FTP server). If not, the connection fails. By default, Enable remote verification check box is selected. This property is applicable to only FTP and FTPS protocols.

### Part 2: Inbound scenario

In this part, you will import the FTP 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 V6.0, 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 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 FTPCOBOLInboundModule
  - \_\_\_\_a. From the Business Integration window, right-click and select New > Module
  - \_\_\_\_b. From the New Module window, enter **FTPCOBOLInboundModule** for the Module Name

🚯 New Module	_ 🗆 X
Create a Module	
Use a module to integrate business applications and services. A module can contain any type component and can be deployed on WebSphere Process Server.	of
Module name: FTPCOBOLInboundModule	
☑ Use default location	
Location: C:\Labfiles70\FTPCobol\workspace\FTPCOBOLInboundModule	Browse
Open module assembly diagram	

- \_\_\_\_ c. Ensure that the box next to **Open module assembly diagram** is checked and then click **Finish**
- \_\_\_\_ d. You will now see a new module, FTPCOBOLInboundModule, created from 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, FTPCOBOLInboundModule, and select **New > Business Object from External Data**

\_\_\_\_b. From the Business Object window, select Languages > COBOL

🚯 Business Object From External Data	_ 🗆 🗙
Input Sources For Creating Business Objects	
Select an input source for creating business objects.	
Eilter: type filter text	E.
Input Sources:	
C (CICS Channel)	
← C (Multiple Output) ← COBOL	
COBOL (CICS Channel)	•
Description:	
Create a new business object by importing data structures from a COBOL file.	<b>A</b>

- \_\_\_ c. Click Next
- \_\_\_\_d. In the Business Object Mapping Details window,
  - 1) Notice that the Select mapping value is COBOL to Business Object
  - 2) Click Browse... and select the traderc99FTP.ccp file located in <FTPFILES>

🌐 New Business Object From External Data	
Provide Details for the Mapping	0
Selected mapping: COBOL to Business Object	<b>T</b>
COBOL file:* C:\Labfiles70\FTPFiles\taderc99FTP.ccp <	Browse

3) Click Next

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

🚯 External (	Data	×
Select Da	ta Structures	
Specify the se data structure	attings used to compile the COBOL, C or PL/I file. Then click Find to discover the es in the file and select from which to build business objects.	$\bigcirc$
		T
Platform:	Win32	
Code page:	ISO-8859-1	Select
Advanced >	>	
Data structure	is:	
DFH	COMMAREA	Find

- \_\_\_\_f. Select DFHCOMMAREA and click Next
- \_\_\_\_g. In the Generate Business Objects screen, ensure that the selected Module is FTPCOBOLInboundModule and click **Finish**

Module:	FTPCOBOLInboundModule	New
Namespace;	http://FTPCOBOLInboundModule	
	🔽 Use default namespace	
Folder:		Browse
Name: *	DFHCOMMAREA	
Generation Style:	Default	

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

🖃 🔁 FTPCOBOLInboundModule
- 😚 Assembly Diagram
- 🗁 Integration Logic
🖻 🗁 🗁 Data Types
🛄 DFHCOMMAREA
🛄 Interfaces
🎰 📥 Mapping

- \_ 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

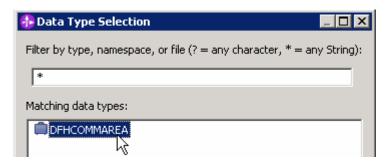
\_\_\_\_b. From the External Data window, select **Adapters > FTP** under Available Types:

🚯 Business Object From External Data	_ 🗆 🗙
Input Sources For Creating Business Objects	
Select an input source for creating business objects.	
	-
Eilter: type filter text	Ē.
Input Sources:	
EMail	
Flat File	
ftp FTP	
🖻 📲 Languages	
⊂	-
Description:	
Create a new business object using the FTP resource adapter to wrap an existing business object and dynamically specify interaction values.	<b>A</b>

\_\_\_\_ c. From the next screen, ensure that the correct module, **FTPCOBOLInboundModule**, is selected and click **Next** 

**Note**: The resource adapter archive file is imported and a new connector project, **CWYFT\_FTPFile**, 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



### 3) Click OK

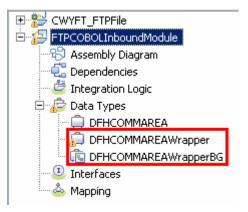
\_\_\_\_e. To generate a business graph, select the Generate business graph for each business object check box

\_\_\_\_\_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://FTPCOBOLInboundModule}	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/ftp	
Generate retrieve container to retrieve multiple 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 FTPCOBOLInboundModule assembly diagram is opened in Assembly editor
  - \_\_\_\_a. In the Business Integration view, expand FTPCOBOLInboundModule 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**:

🕄 FTPCOBOLInboundMo	dule - Assembly Diagram
<ul> <li>✓ <sup>Q</sup> Palette</li> <li>▶ Q Q % </li> <li>▶ Favorites</li> <li>▷ Components</li> </ul>	
C Outbound Adapters	<
™ Email Flat File	
PFF FTP で iSeries うDBC An export compo connects to a rer System (FTP).	nent which note file
DE System (FTP).	

- \_\_\_\_b. Under Inbound Adapters, click the **FTP** and then click the empty canvas of the assembly diagram. The New FTP File Service wizard is opened
- 3. From the New External Service window, expand Adapters > FTP and select Advanced: Create a FTP service using the complete wizard

🤀 New External Service 📃 🗖	X
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:	
☐ S Adapters ☐ □ ftp FTP	
Description:	
Creates a service that reads from or writes to a file on a remote file system using the WebSphere® Adapter for File Transfer Protocol (FTP). You can create integrated processes to access files managed by the FTP server without special coding. During outbound processing, the adapter receives a request from the module in the form of a business object and then the adapter processes the request and returns the result when applicable, in a business object. During inbound processing, the adapter polls a file system associated with a FTP server for events at specified intervals. When the adapter detects an event, it requests a copy of the file, converts the file into a business object, and sends it to the consuming service.	4

#### \_\_\_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 > FTP and click Next

\_ 4. On the Select an Adapter screen, select IBM WebSphere Adapter for FTP (IBM : 7.0.0.0\_IF01) > CWYFT\_FTPFile and click Next

🔒 New External Service	
Select an Adapter Select the version of the adapter that you want to use.	0
IBM WebSphere Adapter for FTP (IBM : 7.0.0.0_IF01)	

- \_\_\_\_ 5. Service Configuration Properties:
  - \_\_\_\_a. Deploy connector project: ensure that the default option With module for use by single application is selected
  - \_\_\_\_b. Enter these for FTP system connection information:
    - 1) Host name: **<FTP\_Machine\_Name>** (or IP Address of the machine that has FTP Server), for Ex: wsbeta149.austin.ibm.com

Note: Alternatively you can also define a WebSphere variable for the host name and replace it here.

2) Remote directory: **\${FTP\_EVENT}** 

Note: This is the directory from where adapter gets the event files.

- 3) Local directory: **\${FTP\_LOCAL\_EVENT}**
- 4) Protocol: FTP file transfer protocol
- 5) Port number: **21** (default)

Connection properties		
FTP system connection information		
Host name: * wsbeta159.austin.ibm.com		
Remote directory:* \${FTP_EVENT}		
Local directory: * {FTP_LOCAL_EVENT}	Browse	
Protocol: FTP - file transfer protocol < 💽		
Port number: ->> 21		

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

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

\_\_\_\_\_d. Event 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".

\_\_\_\_e. FTP archiving configuration:

1) Local Archive directory: click Browse... and select \${FTP\_LOCAL\_ARCHIVE}

#### 2) Remote archive directory: **\${FTP\_ARCHIVE}**

<ul> <li>FTP archiving configuration</li> </ul>		
Specify local archive directory to enable archiving on the local system, specify remote archive directory to enable archiving on the remote system.		
Local archive directory:	\${FTP_LOCAL_ARCHIVE} Browse	
File extension for local archive:	original	
Success file extension for local archive:	success	
Failure file extension for local achive:	fail	
Remote archive directory:	\${FTP_ARCHIVE}	
File extension for remote archive:		

- 6. Secure configuration: Refer to 'Install and configure SSH server lab' and 'FTP Adapter SFTP support lab' for more details on this new feature
- 7. Logging and tracing: Refer to the new lab 'Log and confidential trace lab' for more details on this new feature
- 8. For this lab, you are going to use the security properties from the activation specification.
  - \_\_\_\_a. Select the check box next to Using security properties from the activation specification
  - \_\_\_\_b. User name: username using which you connect to your FTP server (for Ex: root)
  - \_\_\_\_ c. **Password**: password for the user to connect to your FTP server

Servi	ce properties					
H	How do you want to specify the security credentials?					
(	🔿 Using an existing JA	AS alias (recommended)				
	A Java Authenticati	on and Authorization Services (JAAS) alias is the preferred method.				
	J2C authentication of	lata entry:				
- [	Using security prope	rties from the activation specification				
	The properties will b	e stored as plain text; no encryption is used.				
	User name:	* root				
	Password:	* *****				
(	Other					
		required or will be handled by the EIS system, or the RAR will be deployed on rity will be specified by the properties in the JNDI lookup name.				

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

\_\_\_\_\_ 10. For Data format options, select Use COBOL, C or PL/I data format from the drop down list

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

11. 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 emitFTPCOBOL operation:

- \_\_\_\_\_ 12. 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

- \_\_\_\_\_ 13. For **Operation name**, enter any name, for Ex: **emitFTPCOBOL**
- 14. For **Input type**, click **Browse** and select **DFHCOMMAREAWrapperBG** from the Data Type Selection window

Operation name:*	emitFTPCOBOL		
Specify the operation	on input		
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com/xmlns/prod	Browse	New

\_\_\_\_a. Click Finish. The defined operation, emitFTPCOBOL, is populated under Operations list

peratio	ns:	
···· 🍪	emitFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrapperbg}	Add

- \_\_\_\_b. Click Next from Operations screen
- \_\_\_\_ 15. From Generate Service screen, accept the default value, **FTPExport**, for **Name**

Properties for service			
Module:	FTPCOBOLInboundModule	New	
Namespace;	http://FTPCOBOLInboundModule/FTPExport		
	Vse default namespace		
Name: *	FTPExport		
Description:			

\_\_\_a. Click Finish

0

\_\_\_\_\_16. The Assembly diagram for FTPCOBOLInboundModule is opened with an Export component, **FTPExport**:

8 *FTPCOBOLInboundM	lodule - Assembly Diagram 🔀
👌 🔮 Palette	
ि <b>€ € ि </b> ि	· · · ·
🕞 Favorites	P RP FTPExport

\_\_\_\_\_ 17. 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.

- 1. Open the assembly diagram for FTPCOBOLInboundModule (if it is already not open)
  - \_\_\_\_a. From the business integration view, expand FTPCOBOLInboundModule and double click Assembly diagram
- 2. Drop a Java component to onto the assembly diagram
  - \_\_\_\_a. From the Palette, click Components to expand it
  - \_\_\_\_b. Click **Java** and then click the empty space of FTPCOBOLInboundModule assembly diagram. This will place a new component, **Component1** on the assembly diagram.
- \_\_\_\_ 3. Wire the FTPExport to the Component1
  - \_\_\_\_a. Select the wire (11) icon from the Palette
  - \_\_\_\_b. Click FTPExport and then click Component1 to wire them together
  - \_\_\_\_ c. Select **OK** for the Add Wire pop-up window:
  - \_\_\_\_\_d. From the top of the Palette, click the **Selection Tool** icon ( ) to get back to the normal cursor mode
  - \_\_\_\_e. Right-click the empty space of the Assembly diagram and select Arrange Contents Automatically from the pop-up menu

Your assembly diagram for FTPCOBOLInboundModule will look like this:



- \_\_\_\_f. Right-click Component1 and select Generate Implementation from the pop-up menu
- \_\_\_\_g. On the Generate Implementation panel, select default package, and click OK
- \_\_\_\_h. **Component1Impl.java** is opened in Assembly editor. Scroll down to the method **emitFTPCOBOL** that needs to be implemented and add this code under that method:

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

```
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);
```

- \_\_\_\_ i. Save (Ctrl + S) and close Component1Impl.java
- \_\_\_\_j. Save (Ctrl + S) and close Assembly diagram: FTPCOBOLInboundModule

# 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 FTPCOBOLInboundModuleApp under Available projects panel and click Add >
  - \_\_\_\_ c. You will now see the FTPCOBOLInboundModuleApp 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. Test the application by providing input files:

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

- \_\_\_\_a. On the machine where the FTP Server is running, put **COBOLTest.txt** file in the **EventDir**. The adapter will poll the copied file from the event directory and will transfer it to the archive directory
- \_\_\_ b. The file will pass through the only existing method, emitFTPCOBOL, and you should see this message in your Server Logs view (or SystemOut.log):

Туре	▲ Time	Thread ID	Contents
🗖 📑 Log message	Dec 17, 2008 14:05:15.484	000006fe	**************************************
🔲 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	File Name : COBOLTest.txt
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	NAME> FTP11
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	First Name> Joe
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	Last Name> Sean
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	Street> Burnet Rd
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	City> Austin
🔲 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	Country> USA
🔲 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	Phone> 5128380000
🗖 🗐 Log message	Dec 17, 2008 14:05:15.484	000006fe	Postal Code> 78758

#### Verify the results:

- \_\_\_ c. Check the <LOCAL\_EVENT\_DIR> on your local machine. The file is quickly moved from this directory to the local archive directory
- \_\_\_\_\_d. Check the **ArchiveDir** of your FTP server which should contain the same file name appended with year, month, date, system time, and processed as you have given

\_\_\_\_e. Check the **<LOCAL\_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\FTPCobol\LocalArchiveDir				
Folders ×		Name 🔺	Size	Туре	
	🖃 🧰 FTPCobol 📃	COBOLTest.txt_2009_12_08_19_56_29_843.success	1 KB	SUCCESS File	

- \_\_\_\_\_ 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 FTPCOBOLInboundModuleApp 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 FTPCOBOLOutboundModule
  - \_\_\_\_a. From the Business Integration window, right-click and select New > Module
  - \_\_\_\_b. From the New Module window, enter FTPCOBOLOutboundModule 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: FTPCOBOLOutboundModule	
☑ Use default location	
Location: C:\Labfiles70\FTPCobol\workspace\FTPCOBOLOutboundModule	Browse
Open module assembly diagram	

\_\_\_\_ d. Click Finish

You will now see a new module, FTPCOBOLOutboundModule, 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 **FTPCOBOLOutboundModule**
- \_\_\_\_ 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 Object From External Data
  - \_\_\_\_b. From the External Data window, select **Adapters > FTP** under Available Types:
  - \_\_\_\_ c. From the next screen, ensure that the correct module, FTPCOBOLOutboundModule, 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://FTPCOBOLOutboundModule}	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/ftp	
Generate retrieve container to retrieve multiple business objects.	

- \_\_\_\_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.

🖻 🔁 FTPCOBOLOutboundModule
- 😌 Assembly Diagram
- 🐸 Integration Logic
🖻 🕼 Data Types
DFHCOMMAREA
🕞 DFHCOMMAREARetrieveWrapper
🕂 🗓 DFHCOMMAREARetrieveWrapperBG
🔂 DFHCOMMAREAWrapperBG
🖳 🛄 Interfaces
🔤 📥 Mapping

# 3.2. Configure outbound using external service wizard

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. To start the external service wizard from the Palette:
  - \_\_\_\_a. From the **Palette** on the left side of Assembly Diagram, click **Outbound Adapters**:
  - \_\_\_\_b. Under Outbound Adapters, click the **FTP** and then click the empty canvas of the assembly diagram. The New FTP Service wizard is opened
- 2. From the New External Service window, expand Adapters > FTP and select Advanced: Create a FTP 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	Ba
<u>A</u> vailable types:	
Adapters     Adapters     Advanced: Create a FTP service using the complete wizard     Simple: Create an outbound FTP service to write to a remote file	
Description: Creates a service that reads from or writes to a file on a remote file system using the WebSphere® Adapter for File Transfer Protocol (FTP). You can create integrated processes to access files managed by the FTP server without special coding. During outbound processing, the adapter receives a request from the module in the form of a business object and then the adapter processes the request and returns the result when applicable, in a business object. During inbound processing, the adapter polls a file system associated with a FTP server for events at specified intervals. When the adapter detects an event, it requests a copy of the file, converts the file into a business object, and sends it to the consuming service.	4

\_\_a. Click Next

Note: You can also start the external service wizard 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 > FTP and click Next

3. On the Select an Adapter screen, expand IBM WebSphere Adapter for FTP (IBM : 7.0.0.0\_IF01) and select CWYFT\_FTPFile



- \_\_\_a. Click Next
- 4. Service Configuration Properties:
  - \_\_\_\_a. Deploy connector project: ensure that the default option With module for use by single application is selected
  - \_\_\_\_b. Enter these for FTP system connection information:
    - 1) Host name: **<FTP\_Machine\_Name>** (or IP Address of the machine that has FTP Server), for Ex: wsbeta159.austin.ibm.com
    - 2) Directory: **\${FTP\_OUT}**

**Note**: This is the folder where the adapter will create the file.

#### 3) Protocol: FTP – file transfer protocol

4) Port number: **21** (default)

Deploy connector project:	With module for use by single application 🔫 🚽 🔽
Connection properties:	Use properties below
Connection properties	
FTP system connection informat	ion
Host name: * wsbeta159.	austin.ibm.com 🔫 ——
Directory: * \${FTP_OUT	
Protocol: FTP - file tra	nsfer protocol <del> –</del>
Port number: 21-	

- 5. Optionally, click **Advanced >>** to see the hidden advanced properties that can be configured:
  - \_\_\_\_a. Secure configuration: Refer to the 'Install and configure SSH server lab' and 'FTP Adapter SFTP support lab' for more details on this new feature
  - \_\_\_\_b. Logging and tracing: Refer to the new lab 'Log and confidential trace lab' for more details on this new feature
- \_\_\_\_6. For this lab, you are going to use the security properties from the activation specification.
  - \_\_\_\_a. Select the check box next to Using security properties from the activation specification
  - \_\_\_\_b. User name: username using which you connect to your FTP server (for Ex: root)

\_\_\_\_ c. **Password**: password for the user to connect to your FTP server

Service proj	perties			
How do	o you want to specify the security o	redentials?		
🔿 Usir	ng an existing JAAS alias (recomme	nded)		
A J	A Java Authentication and Authorization Services (JAAS) alias is the preferred method.			
J20	C authentication data entry:			
🖲 Usir	ng security properties from the acti	vation specification		
The	e properties will be stored as plain (	ext; no encryption is used.		
Use	er name: * (roo	t		
Pas	ssword: * 💌	*****		
O Oth	ner			
		handled by the EIS system, or the RAR will b ed by the properties in the JNDI lookup name.		

\_\_\_\_\_7. For Data format options, select Use COBOL, C or PL/I data format from the drop down list

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

8. 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 createFTPCOBOL operation:

- 9. From the Operations screen, click Add...
  - \_\_\_\_a. For **Operation kind**, select **Create** from the drop down list
  - \_\_\_\_b. For Data type for the operation input, select User defined type from the drop down list
  - \_\_\_\_ c. Select the check box next to Enable response type for the operation

🚯 Add Operation		_ 🗆 🗵		
<b>Operation</b> Specify the properties for the operation	5			
Operation kind:	Create	<b>~~</b> •		
Operation properties				
Data type for the operation:	User defined type	<b>←</b>		
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 any name, for Ex: **createFTPCOBOL** 

\_\_\_\_f. For Input type, click Browse and select DFHCOMMAREAWrapperBG from the Data Type Selection window

Operation name: *	createFTPCOBOL				
Specify the operation	on input				
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com/	Browse	New		
Specify the operation output					
Output type;	CreateResponse {http://www.ibm.com/xmlns/prod/w	Browse,	New		

\_\_\_\_g. Click Finish. The defined operation, createFTPCOBOL, is populated under Operations list

¢	Operatio	ns:	
	···· 🔅	createFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrapperb	Add

#### Define appendFTPCOBOL operation:

\_\_\_\_\_10. 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 and click Next
- \_\_\_\_ d. For Operation name, enter appendFTPCOBOL
- \_\_\_\_ e. For Input type, click Browse and select DFHCOMMAREAWrapperBG from the Data Type Selection window

Operation name: *				
Specify the operation input				
Input type: *	DFHCOMMAREAWrapperBG {http://www.ibm.com	Browse	New	
Specify the operation output				
Output type;	AppendResponse {http://www.ibm.com/xmlns/prod/websp	Browse,	New	

\_\_\_\_f. Click Finish. The defined operation, appendFTPCOBOL, is populated under Operations list

Operations:	
createFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrappe	Add
appendFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrapp	Edit
	Remove
Operation properties:	
InteractionSpec properties for 'appendFTPCOBOL'	
FTP system connection information	
Remote directory on FTP system:	
Default target file name:	
File in local directory	
Local directory: Bro	wse,,,
Archive file in the local directory for create operation	
Local archive directory for create operation: Bro	wse,,,
Create new file if the file does not exist	
🔲 Generate a unique file	
Delete the file after retrieve operation	
Remote archive directory for retrieve operation:	
Advanced >>	

#### Define retrieveFTPCOBOL operation:

- \_\_\_\_\_ 11. 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
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 **FTPFile** (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 retrieveFTPCOBOL
- \_\_\_\_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):				
*				
<u> </u>				
Matching data types:				
DFHCOMMAREAWrapper				

#### 3) Click OK

\_\_\_\_g. You should now see this:

Operation name: *	retrieveFTPCOBOL				
Specify the operation input					
Input type;	FTPFile {http://www.ibm.com/xmlns/prod/websphere/j	Browse,	New		
Specify the operation output					
Output type:*	DFHCOMMAREARetrieveWrapperBG {http://www.#5.	Browse	New		

- \_\_\_h. Click **Finish** from the Add Operation window. The operation, retrieveFTPCOBOL, will now be displayed under Operations list.
- \_\_\_\_\_i. You can click **Advanced >>** under 'InteractionSpec properties for retrieveCOBOL' to review the properties available at Interaction spec level

Operations:	
<ul> <li>createFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrappe</li> <li>appendFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/dfhcommareawrappe</li> <li>retrieveFTPCOBOL ({http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/ftpfile}FTPFile) : {h</li> </ul>	
Operation properties:	
InteractionSpec properties for 'retrieveFTPCOBOL'	
FTP system connection information	
Remote directory on FTP system:	
Default target file name:	
E File in local directory	
Local directory: Br	rowse
Archive file in the local directory for create operation	
Local archive directory for create operation:	rowse,,,
Create new file if the file does not exist	
🔽 Generate a unique file	
Delete the file after retrieve operation	
Remote archive directory for retrieve operation:	
Advanced >>	

- \_\_\_j. Click Next from Operations screen
- \_\_\_\_\_ 12. From Generate Service screen, accept the default value, FTPImport, for Name

Properties for service					
Module:	FTPCOBOLOutboundModule	New			
Namespace;	http://FTPCOBOLOutboundModule/FTPImport				
	✓ Use default namespace				
Name: *	FTPImport				
	Save business objects to a library				
Library:	<b>_</b>	New			
Description:					

\_\_\_a. Click Finish

\_\_\_\_\_ 13. The Assembly diagram for FTPCOBOLOutboundModule is opened with an Import component, **FTPImport**:

*FTPCOBOLOutbound	Module - Assembly Diagram 🕺
👌 🔮 Palette	
<u>ि</u> • • • • •	
🔁 Favorites	ft FTPImport

\_\_\_\_\_ 14. 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. 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 FTPCOBOLOutboundModuleApp under Available projects panel and click Add >
  - \_\_\_\_ c. You will now see the FTPCOBOLOutboundModuleApp 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 **FTPCOBOLOutboundModule** and select **Test > Test Module**
  - \_\_\_\_b. The FTPCOBOLOutboundModule\_Test window is opened in the Assembly editor

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

- createFTPCOBOL
- appendFTPCOBOL
- retrieveFTPCOBOL

#### Test Create operation:

- 3. Under Detailed Properties, for the Operation field, select createFTPCOBOL from the drop down menu
  - 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:	FTPCOBOLOutboundModule	
Component:	FTPImport	
Interface:	FTPImport	
Operation:	createFTPCOBOL	

Fill out the fields for Initial request parameters:

\_\_\_\_a. For Filename, enter any name, for Ex: FTPCOBOLTest.txt

## \_\_\_\_b. For includeEndBODelimiter, enter ###

Name	Туре	Value
🖃 🏪 createFTPCOBOLInput	DFHCOMMAREAWrapperBG	аь
💭 verb	verb <string></string>	I CREATE
🗄 🖳 DFHCOMMAREAWrapper *	DFHCOMMAREAWrapper	аь
İII DirectoryPath	string	аь
- Eilename	string	📧 FTPCOBOLTest.txt
ChunkInfo	string	аь
FtpServerHostName	string	аь
- FtpServerEventDirectory	string	аь
💭 DataConnectionMode	string	аь
🖳 🛄 FileTransferType	string	ab
💭 DataProtectionLevel	string	аь
- EcondServerDirectory	string	аь
💭 SecondServerUsername	string	35
💭 SecondServerPassword	string	35
FileContentEncoding	string	ab
IncludeEndBODelimiter	string	⊡ ###
FileInLocalDirectory	boolean	📧 false

# \_\_\_\_ c. For the fields under **Content**, enter any values:

🗄 📲 Content	DFHCOMMAREA	ab
🎞 CustomerNumber *	CustomerNumber <string></string>	B FTP11
🖳 🛄 FirstName *	FirstName <string></string>	🗈 Joe
師 LastName *	LastName <string></string>	📧 Sean
師 Street *	Street <string></string>	🗈 Burnet Rd
🛄 City *	City <string></string>	📧 Austin
抑 Country *	Country <string></string>	III USA
抑 Phone *	Phone <string></string>	5128380000
🛄 PostalCode *	PostalCode <string></string>	ab 78758

#### \_\_\_\_ d. Click Continue button under Events

#### 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>



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

Deployment Location	_ <b>D</b> X
Select a Deployment Location Specify a runtime location where this test will deploy.	E
Deployment location:	
🖃 🗟 WebSphere Process Servers	New Server
WebSphere Process Server v7.0 at localhost	
Eclipse 1.5 JVM	
WebSphere Enterprise Service Bus Server	
Mode: Run	
$\square$ Use this location as the default and do not ask again	

- \_\_\_\_f. 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). Please sign in to continue the test. <u>U</u> ser ID:
admin
Password:
••••
✓ Use the authentication settings in the preference and <u>n</u> ever ask again.
OK Cancel

- \_\_\_\_ g. Verify the created file and its contents:
  - 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 event to display its properties in the General Properties and Detailed Properties sections. More	▼ Detailed Properties				
	Module:       FTPCOBOLOutboundModule         Component:       FTPImport         Interface:       FTPImport         Operation:       createFTPCOBOL         Return parameters:       Component				
Invoke returned	Value Editor XML Source	Type CreateResponse string	Value		

2) You will see a new file, **FTPCOBOLTest.txt** created under **OutDir** on your FTP server. Open that file to verify the content

## Test Append operation:

- 4. Under **Detailed Properties**, note that the Operation is **appendFTPCOBOL** 
  - \_\_\_\_a. Fill out the fields for Initial request parameters:
  - \_\_\_\_b. For fileName, enter FTPCOBOLTest.txt
  - \_\_\_ c. For includeEndBODelimiter, enter ###

Name	Туре	Value
🖃 🏪 appendFTPCOBOLInput	DFHCOMMAREAWrapperBG	аь
···· Ū⊐ verb	verb <string></string>	IB CREATE
🗄 🖳 DFHCOMMAREAWrapper *	DFHCOMMAREAWrapper	ab
- DirectoryPath	string	ар
- Eilename	string	Im FTPCOBOLTest.txt
💭 ChunkInfo	string	35
- EtpServerHostName	string	ab
- EtpServerEventDirectory	string	ab
💭 DataConnectionMode	string	ab
- 💭 FileTransferType	string	ab
- IntaProtectionLevel	string	ab
- EcondServerDirectory	string	ab
- 💭 SecondServerUsername	string	ab
- EcondServerPassword	string	ар
- EileContentEncoding	string	ар
IncludeEndBODelimiter	string	<b>35 ###</b>

\_\_\_\_\_d. Enter the values of your choice for the fields under Content

🗄 🖳 📴 Content	DFHCOMMAREA	ab
- T CustomerNumber *	CustomerNumber <string></string>	🗈 XYZ9
抑 FirstName *	FirstName <string></string>	🗈 James
🎞 LastName *	LastName <string></string>	🗈 Doe
抑 Street *	Street <string></string>	📧 Duval Rd
抑 City *	City <string></string>	🗈 Austin
抑 Country *	Country <string></string>	III USA
抑 Phone *	Phone <string></string>	5128380000
🛄 PostalCode *	PostalCode <string></string>	ab 78758

#### \_\_\_\_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					
This area displays the events in a test trace. Select an event to display its		<ul> <li>Detailed Properties</li> </ul>				
properties in the General Properties and Detailed Properties sections. More		Modi	ular	FTPCOBOLOutboundModu	de	
▶ ■ ↓ ▼   ■ ▼   □ ↓   □		_		FTPImport	<u>ne</u>	
With the second se			_	FTPImport		
Invoke started Invoke (FTPImport:createFTPCOBOL)		<u>O</u> per	ration:	appendFTPCOBOL		
Return (FTPImport:createFTPCOBOL)		Retur	n param	eters:		
Invoke returned		Value	e Editor	XML Source		
Invoke (FTPImport:appendFTPCOBOL)     R    Nvoke started		E	86	Ē		
Invoke (FTPImport:appendFTPCOBOL)				Name	Туре	Value
Return (FTPImport:appendFTPCOBOL)			⊡ 💾	appendFTPCOBOLOutput	AppendResponse	аь
Invoke returned	1		i	💭 Filename	string	B FTPCOBOLTest.txt

\_\_\_\_g. To verify your test results, open the FTPCOBOLTest.txt file from **OutputDir** of your FTP server 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

FTP11Joe	Sean	Burnet Rd	Austin	USA	5128380000	78758	###
XYZ9 James	Doe	Duval Rd	Austin	USA	5128380000	78758	###

#### Test Retrieve operation:

5. Under **Detailed Properties**, note that the Operation is **retrieveCOBOL** 

- \_\_\_\_a. Fill out the fields for Initial request parameters:
- \_\_\_\_b. For filename, enter FTPCOBOLTest.txt
- \_\_\_\_ c. LocalDirectoryPath: provide absolute path to any local directory
- \_\_\_\_d. splitFunctionClassName: com.ibm.j2ca.utils.filesplit.SplitByDelimiter

# \_\_\_e. splitCriteria: ###

Name	Туре	Value
🖃 🏪 retrieveFTPCOBOLInput	FTPFile	ав
🖳 DirectoryPath	string	ав
	string	Im FTPCOBOLTest.txt
💭 ChunkInfo	string	ав
- ImpServerHostName	string	ав
- EtpServerEventDirectory	string	ав
🂭 DataConnectionMode	string	аь
🂭 FileTransferType	string	аь
💭 DataProtectionLevel	string	ав
💭 SecondServerDirectory	string	ав
- I SecondServerUsername	string	ав
💭 SecondServerPassword	string	ав
抑 FileContentEncoding	string	ав
🎞 IncludeEndBODelimiter	string	ав
- IleInLocalDirectory	boolean	🕮 false
	string	💷 C:\Labfiles70\FTPCobol\LocalEventDir
LocalArchivingEnabledForCreate	boolean	🕮 false
- IocalArchiveDirForCreate	string	ав
- Implimited a staging Directory	string	ав
🎞 GenerateUniqueFile	boolean	💷 false
	boolean	🙃 false
[I] ScriptFileParameters	string[]	667
	string	🕮 com.ibm.j2ca.utils.filesplit.SplitByDelimiter
	string	Image: a transformed and t
🎾 DeleteOnRetrieve	boolean	ab false

\_\_\_f. Click Continue button under Events

# \_\_\_\_ g. Verify the results:

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

Name	Туре	Value
🖂 🏪 retrieveFTPCOBOLOutput	DFHCOMMAREARetrieveWrapperBG	аь
····· 💭 verb	verb <string></string>	86
🗄 🖳 DFHCOMMAREARetrieveWrapper *	DFHCOMMAREARetrieveWrapper	аы
⊡…[□] Content	DFHCOMMAREA[]	ഒറ
🚊 🏪 Content[0] <del></del>	DFHCOMMAREA	аы
CustomerNumber *	CustomerNumber <string></string>	ED FTP11
💭 FirstName *	FirstName <string></string>	De Joe
💭 LastName *	LastName <string></string>	💷 Sean
💭 Street *	Street <string></string>	💷 Burnet Rd
🛄 City *	City <string></string>	ab Austin
Country *	Country <string></string>	LED USA
Phone *	Phone <string></string>	□□□ 5128380000
Environment PostalCode *	PostalCode <string></string>	78758
🖻 🖳 Content[1] 🔫	DFHCOMMAREA	аы
CustomerNumber *	CustomerNumber <string></string>	💷 XYZ9
FirstName *	FirstName <string></string>	🗈 James
🛄 LastName *	LastName <string></string>	Doe Doe
💭 Street *	Street <string></string>	💷 Duval Rd
(İll *	City <string></string>	ab Austin
Country *	Country <string></string>	IBD USA
抑 Phone *	Phone <string></string>	■ 5128380000
PostalCode *	PostalCode <string></string>	<b>ab</b> 78758

- \_\_\_\_\_6. 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 FTPCOBOLOutboundModuleApp 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

# **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 FTPCOBOL\_PI.zip from <LAB\_FILES>\FTPCobol\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.	Ľ
Select an import source:	_
Test Web Web services To Compare XML Compare And Andel Project Interchange	
⑦ < Back Next > Einish	Cancel

- \_\_\_\_d. For the From zip file, click on the Browse button and select the FTPCOBOL\_PI.zip in the <LAB\_FILES>>\ FTPCobol\solution directory
- \_\_\_\_e. Enter <LAB\_FILES>\FTPCobol\workspace for the Project location root

\_\_\_\_f. Click the **Select All** button. This will select all the modules: **CWYFT\_FTPFile**, **FTPCOBOLInboundModule**, and **FTPCOBOLOutboundModule** 

🌐 Import Project Interchange Contents	
Import Projects Import Projects from a zip file.	ļ,
From zip file:       C:\Labfiles70\FTPCobol\solution\FTPCOBOL_PI.zip         Project location root:       C:\Labfiles70\FTPCobol\workspace	Browse
CWYFT_FTPFile  CHYPET_FTPCOBOLInboundModule  FTPCOBOLOutboundModule	
Select All Select Referenced	
⊘     < Back     Mext >	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 FTP 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
Image: Service projects on the WebSphere Process Server v7.0.
Ruis 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:						
Fronje name,	Configure profiles			<u>/////////////////////////////////////</u>		
Server connection types a	Server connection types and administrative ports					
C Automatically determine	ne connection s	ettings				
Manually provide conn	ection settings	]				
Connection Type	Port	Default port	Description			
IPC		9633	Recommended for local servers			
RMI		2809	Designed to improve communication w			
SOAP	8880	8880	Designed to be more firewall compatib	ble		
Run server with resource	es within the w	vorkspace				
Security is enabled on t						
Current active authent						
User ID:	ssadmin	<i>.</i>				
USEL ID.	ssaumin					
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