# Log and confidential trace lab

What this exercise is about
Lab requirements
What you should be able to do1
Introduction2
Exercise instructions
Part 1: Initialize the workspace and prepare for the lab5
Part 2: Configure non-pass through inbound scenario6
2.1. Add Java component
Part 3: Deploy the application and enable tracing
Part 4: Load trace file and review confidential messages25
What you did in this exercise
Task: Adding 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 some of new features in WebSphere Adapter for Flat Files V6.2 – usage of Adapter ID for logging and tracing details, hide confidential trace, and poll sub directories in event directory.

# Lab requirements

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

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

# What you should be able to do

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

- Import Flat File adapter RAR file into WebSphere Integration Developer
- Use External Service wizard to configure Activation Spec Properties, Resource Adapter Properties to generate Business Objects
- Deploy the adapter application onto WebSphere Process Server and enable tracing in WebSphere Process Server administrative console

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

# Introduction

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

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

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

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

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

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

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

# **Exercise instructions**

Some instructions in this lab are Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to run the appropriate commands, and use 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\WID62	
<wps_home></wps_home>	C:\ <wid_home>\runtimes/bi_v62</wid_home>	
<ffadapter_home></ffadapter_home>	<wid_home>\ResourceAdapters\FlatFile_6.2.0.0\deploy</wid_home>	
<lab_files></lab_files>	C:\Labfiles62	/tmp/Labfiles62
<workspace></workspace>	<lab_files>\LogAndTrace\workspace</lab_files>	
<event_dir></event_dir>	<lab_files>\LogAndTrace\eventdir</lab_files>	
<archive_dir></archive_dir>	<lab_files>\LogAndTrace\archivedir</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:\Labfiles62\ with C:/Labfiles62/

### Instructions if using a remote server for testing

Note that the previous table is relative to where you are running WebSphere Integration Developer. The 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 "<u>Task: Adding remote server to WebSphere Integration Developer</u> test environment".

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

This part of the lab, you will start the WebSphere Integration Developer V6.2 with a new workspace and create required data source and database using the administrative console of WebSphere Process Server V6.2

- 1. Extract the provided Labfiles62.zip to your C:\ (root) drive, if you have not already done so. This will create the necessary subdirectory structure to complete the lab, and provides you with sample text files
- 2. Start the WebSphere Integration Developer V6.2 with a new workspace
  - \_\_\_\_a. Select Start > All Programs > IBM WebSphere Integration Developer > IBM WebSphere Integration Developer V6.2 > WebSphere Integration Developer V6.2
  - \_\_\_\_b. From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field

🚯 Workspace Launcher			×
Select a workspace			
IBM WebSphere Integration Developer 6.2 stores your projects in a folder called a work Choose a workspace folder to use for this session.	space.		
			[]
Workspace: C:\Labfiles62\LogAndTrace\workspace		-	Browse
▶ <u>C</u> opy Settings			
0	ОК		Cancel

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

Name	Value
LOGTRACE_EVENT	C:\Labfiles62\LogAndTrace\eventdir
LOGTRACE_ARCHIVE	C:\Labfiles62\LogAndTrace\archivedir

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

# Part 2: Configure non-pass through inbound scenario

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

- \_\_\_\_\_1. Create the module: LogAndTraceModule
  - \_\_\_\_a. From the Business Integration window, right-click and select New > Module

\_\_\_\_b. From the New Module window, enter LogAndTraceModule for the Module Name

Rew Module		
łodule		
reate a new business integration module. A module is a project that is use ersion management, organizing resources, and deploying to the runtime er	d for development, nvironment.	
Module Marile, TrogAnd Hacemodule		
Use default location		
Location: C:/Labfiles62/LogAndTrace/workspace/LogAndTraceModule		Browse
Open module assembly diagram		

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

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

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

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

- \_\_\_\_ d. Select the box next to Customer.xsd
- \_\_\_\_e. Ensure that the LogAndTraceModule is selected for Into folder

\_\_\_\_f. Click Finish from the Import window

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

- \_\_\_\_ 3. Review imported business objects:
  - \_\_\_\_a. Expand LogAndTraceModule > Data Types and you will now see two new data types Customer and Order under it.
  - \_\_\_\_b. Double-click **Customer** review the fields inside the object:

[	📋 Customer	7
	<u> </u>	
	customerName	string
	e Address	string
	e City	string
	e State	string
,		

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

🕄 FFPSInboundModule -	Assembly Diagram	ß
👌 😳 Palette		
\$®,€,€, <b>®</b> ,		
🔁 Favorites		
🔁 Components		
🔁 Outbound Adapters		
🕞 Inbound Adapters 🛛 🖣	<	
🛃 Email		
Flat File		
ff <sup>#</sup> FTP 🔓		
iSeries An export com	ponent which	
	ocarnie system.	
J JD Edwards Enter		
PeopleSort		
<b>1</b> 5,000		

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

\_\_\_\_\_7. From the Flat File Service screen, select **Create a service (advanced)** 

🚯 New Flat File Service 🔀
Flat File Service
Create a new flat file service.
Create a service from a pattern (typical) Create a service (advanced)
<u>A</u> vailable Patterns:
E 🔤 Integration
È
Hat File Create an inhound Elat File service to read from a local file
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.
② < <u>B</u> ack <u>N</u> ext > Einish Cancel

#### \_\_\_a. Click Next

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

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

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

\_ 8. On the Select an Adapter screen, select IBM WebSphere Adapter for Flat Files (IBM : 6.2.0.0) and click Next

🚯 External Service	×
Select an Adapter Select the adapter you want to use.	0
IBM WebSphere Adapter for Flat Files (IBM : 6.2.0.0)	

9. Adapter Import screen:

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

- \_\_\_\_a. The default Connector file is selected which is shipped along with WebSphere Integration Developer
- \_\_\_\_b. Accept the default name for Connector project, **CWYFF\_FlatFile**. You can change it to any other name, but for this lab, you can leave the default name
- \_\_\_\_ c. For Target server, ensure that WebSphere Process Server v6.2 is selected

🚯 External Service		_ 🗆 🗙
Adapter Import		
Import a resource ad project for the adapt	apter archive (RAR) from the file system to create a connector er.	
Archive file:	C:\IBM\WID62\ResourceAdapters\FlatFile_6.2.0.0\deploy\CWYFl	F_FlatFile.rar
Connector project:*	CWYFF_FlatFile	
Target runtime:	WebSphere Process Server v6.2	

\_\_\_\_d. Click Next

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

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

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

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

Deploy connector project:	With module for use by single application	•	
Connection properties:	Use properties below	~	
Connection properties			
File system connection informat	ion		
Event directory:* \${LOGTRAC	E_EVENT}		Browse

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

\_\_\_\_ 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 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:

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

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

\_\_\_\_g. Advanced properties:

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

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

<ul> <li>Advanced Properties</li> </ul>	
Retrieve files with pattern: * *.*	
Pass only file name and directory, not the content	
$\square$ Include business object delimiter in the file content	
Retrieve files in sorted order: No sort	
File content encoding: UTF-8	Select
Specify the splitting function class name and the split criteria to split the file co	ntent
Split file content based on size (bytes) or delimiter	
Specify criteria to split file content: 0	
Split function class name: com.ibm.j2ca.utils.filesplit.SplitBySize	Browse
Poll subdirectories in event directory	

\_\_\_\_h. File archiving configuration:

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

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

👻 File ar	hiving configuration		
Specif	y archive directory to archive	processed inbound event files	
Archiv	e directory:	\${LOGTRACE_ARCHIVE}	Browse
File e>	tension for archive:	original	
Succe	ss file extension for archive:	success	
Failure	e file extension for archive:	fail	

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

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

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

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

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

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

<ul> <li>Logging and tracing</li> </ul>	
Adapter ID:* 001	
Disguise user data as "XXX" in log and trace files.	

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

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

\_\_\_\_a. Click Select next to Function selector. The Binding Resource Configuration window is opened

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

🚯 Function Selector Configuration	_ 🗆 🗙
Select Function Selector Select a function selector entry from the list. If you want to use your own custom function selector then select the second radio button to add your custom function selector.	٥
• Use existing function selector from the list	
	<b>#</b>
EmbeddedNameFunctionSelector properties	
Select your custom function selector from the workspace  Function selector class name:	Select
Add custom class to binding registry	
Description:	

#### \_\_\_ c. Click Next

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

#### Data handler configuration:

\_\_\_\_\_d. Click **Select...** next to Data handler configuration name. A Binding Resource Configuration window is opened for you to define the data handler

# \_\_\_\_ e. Under 'Use existing data format transformation from the list', select XML > UTF8XMLDataHandler

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

🚯 Data Handler Configuration 📃 🗖 🗵
Select Data Format Transformation Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.
Use existing data format transformation from the list
<b>⊕</b>
Delimited Fixed width Handled by WTX Handled by WTX Invoker JAXB based Java bean JSON JSON UTF8XMLDataHandler
Select your custom data format transformation from the workspace
Data transformation class name: Select
Add gustom class to binding registry
Description:
On inbound, parses UTF-8 encoded XML data into a business object. On outbound, serializes business object to UTF-8 encoded XML data.

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



\_\_\_h. Click Next

\_\_\_\_\_i. From the New Function Selector Configuration screen, enter CustomFS for Name

Mod <u>u</u> le or library:	LogAndTraceModule	Browse	N <u>e</u> w
Na <u>m</u> espace:	http://LogAndTraceModule	🔽 Default	
F <u>o</u> lder:		Bro <u>w</u> se	
N <u>a</u> me:	CustomFS		

#### \_\_\_j. Click Finish

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

Service properties			
Function selector options:	Use a function selector configuration	<▼	
Function selector: *	CustomFS	٠	Select

#### Data binding configuration:

- 12. You can define data binding in two places service level (current screen of External Service wizard) or later at the method level (Operations screen of the External Service wizard). In this lab, you will define data binding at the service level (from this screen)
  - \_\_\_\_a. From the dropdown menu next to Data format options, select 'Use a data binding configuration for all operations'

Data format options:	Use a data format configuration for all operations	
Data format:	Not defined	Select

\_\_\_\_\_b. Click **Select...** next to **Data format**. A Binding Resource Configuration window is opened

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

🚯 Data Binding Configuration	
Select Data Format Transformation	
Select a data format transformation entry from the list. If you want to use your own custom data transformation then select the second radio button to add your custom transformation.	
Use existing data format transformation from the list	
	<b>#</b>

- \_\_\_ d. Click Next
- \_\_\_\_e. Select the UTF8XMLDataHandler for data handler:
  - 1) Click Select next to 'Configured data handler'

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

🚯 Data Binding Configu	ration	
Data Transformatio	n Properties	
Specify the properties for	the data transformation.	
Select DataBinding if you adapter.	want to use a data binding developed for earlier version:	s of the
Binding type:	DataHandler 🔨 🔫	
Configured data handler:	UTF8XMLDataHandler	<u>S</u> elect
Configured data binding;	Not defined	Select

- \_\_\_h. Click Next
- \_\_\_\_i. Note that the selected module is LogAndTraceModule

1) For the Name, enter CustomDB

👍 Binding Resource Configuration 📃 🗖 🗙		
New Data T	ransformation Configuration	
Create a new o name for the d	data transformation configuration. Specify the module, folder, namespace, and Long transformation configuration.	
<u>M</u> odule:	LogAndTraceModule Browse New	
Name <u>s</u> pace:	http://LogAndTraceModule	
F <u>o</u> lder:	Browse	
N <u>a</u> me:	CustomDB	

#### 2) Click Finish

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

Ser	vice properties		
Dei	vice propercies		
	Function selector option	s: Use a function selector configuration	<b>•</b>
	Function selector:	* CustomFS	Select
	Data format options:	Use a data format configuration for all operations	<b>←</b> ▼
	Data format:	* CustomDB	Select

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

#### Define emitCustomerFile operation:

- \_\_\_\_\_14. From the Operations screen, click **Add...**. Add Operation window is opened
  - \_\_\_\_a. Select User defined type for the Data type and click Next

🚯 Add Operation		×
Operation Select the operation	kind.	5
Operation properti Data type for t	es	
0	< Back Next > Finis	sh Cancel

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

- \_\_\_\_b. For Operation name, enter emitCustomerFile
- \_\_\_\_ c. Define Input type:
  - 1) Click **New...** next to **Input type** to open a New Business Object window
  - 2) From this window, ensure that the Module selected is LogAndTraceModule and click Next
  - 3) Click Browse... next to Data type
  - 4) From the Data Type Selection window, select **Customer** under Matching data types:

Matching data types:			
Customer Order			

- 5) Click OK
- 6) From the wizard, check the box next to Generate business graph for each business object

7) Do not check the box for 'Generate retrieve container to retrieve multiple business objects'

🤂 New Business Object	
Business Object Properties	
Specify the properties for the new business object.	
Data type:  * Customer {http://www.ibm.com/xmlns/prod/websphere	Browse New
Namespace for generated business graph and container object.	
Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile	
🦳 Generate retrieve container to retrieve multiple business objects. 🔫 🗕 🗕	

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

🚯 Add Operation	
<b>Operation</b> Specify the properties for the operation to add.	
Operation name: * emitCustomerFile Specify the operation input	
Input type: * CustomerWrapperBG {http://ww	ww.ibm.com/xmlns/proc
Data format options: Use data format configuration 'C	ustomDB' <del>&lt;</del>
Data format: Not defined	Select

### \_\_\_ f. Click Finish

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

Operations:			
emitCustomerFile ({http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile/customerwrapperbg}CustomerWrapperBG): void	Add		
h. Click <b>Next</b>			

- \_\_\_\_\_15. From the Generate Artifacts screen, enter these:
  - \_\_\_\_a. For Name, enter LogAndTraceInterface

🚯 External Servi	ce	×
Generate Ser Specify the name a	<b>vice</b> Ind location of the new service and its interface.	
Properties for service	vice	
Module:	LogAndTraceModule	▼ New
Namespace;	http://LogAndTraceModule/LogAndTraceInterface	
	Vise default namespace	
Name: *	LogAndTraceInterface	
Description:		

#### \_\_\_\_b. Click Finish

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

🖏 *LogAndTraceModule - Assembly Diagram 🗙				
Image: Palette         Image: Palette           Image: Palette         Image: Palette         Image: Palette				
	LogAndTraceInterface			
🗁 Components				

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

# 2.1. Add Java component

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

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

👌 😳 Palette		
<b>}®, ©, °⊾ <u>®</u>,</b>		
🔁 Favorites		
🔁 Components		
🖳 Untyped Component		
<u>श</u> ी Human Task		
🚰 Java	J	
🔁 Mediatôn Flow		
👤 Proce <mark>Java</mark>		
E Rule Group		
State Machine		

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

🚯 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?		
🗆 Alw	ays create without prompt		
	OK Cancel		

Your assembly diagram for LogAndTraceModule will look like this:

🐾 *LogAndTraceModule - Assembly Diagram 🗙				
Palette  Palette Palett	DegAndTraceInterface	- Omponent 1		

- \_\_\_\_ d. From the top of the Palette, click the **Selection Tool** icon ( ) to get back to the normal cursor mode
- \_\_\_\_\_ 4. Generate Java Implementation
  - \_\_\_a. Right-click Component1 and select Generate Implementation from the pop-up menu
  - \_\_\_\_b. On the Generate Implementation panel, select default package, and click OK
  - \_\_\_\_ c. Component1Impl.java is opened in Assembly editor. Scroll down to the method emitCustomerFile(DataObject emitCustomerFileInput) that needs to be implemented and add this code under that method:

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

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

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

## Part 3: Deploy the application and enable tracing

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

- 1. Start the server (if not already started)
  - \_\_\_\_a. From the Servers view, right-click the WebSphere Process Server v6.2 and select Start
  - \_\_\_\_b. Wait until the server status turns to Started
  - 2. Add the project to the WebSphere Test Environment server
    - \_\_\_\_a. Right-click **WebSphere Process Server v6.2** under the Servers view and select **Add and remove projects...** from the pop-up menu
    - \_\_\_\_ b. From the Add and Remove Projects window, select LogAndTraceModuleApp under Available projects panel and click Add >
    - \_\_\_\_ c. You will now see the LogAndTraceModuleApp added to the Configured projects
    - \_\_\_\_ d. Click Finish and wait until the project is being published onto the server. The server will start in Debug mode if it is not already started before
- \_\_\_\_ 3. Enable tracing
  - \_\_\_\_a. Right-click WebSphere Process Server v6.2 and select Administration > Run administrative console
  - \_\_\_\_b. From the 'Admin Console' window, enter Administrative User ID and Password and then click Log in



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

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

ogging and Tracing ? _				
Logging and Tracing				
Use this page to specify how the server handles log records. You can select an application server to enable or disable a system log for that server, specify where data is stored, and choose a format for log content. You can also specify a log d level for components and groups of components.				
Server 🛟	Node 🗘	Version 🗘	Туре 🗘	
<u>server1</u>	widNode	Business Process Choreographer 6.1.2.0 ND 6.1.0.17 WPS 6.1.2.0	servers	
Total 1				

\_\_\_\_e. Click Change Log Detail Levels under General Properties

Logging and Tracing ?	-
Logging and Tracing > server1	
Use this page to select a system log to configure, or to specify a log detail level for components and groups of components. Use log levels to control which events are processed by Java logging.	
General Properties	_
Diagnostic Trace	
JVM Logs	
Process Logs	
IBM Service Logs	
Change Log Detail Levels	

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

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

Logging and Tracing > see	rver1 > Change Log Detail Levels
Use log levels to control v specify a log detail level f for a predefined group of level. Log detail levels an subsequent levels.	which events are processed by Java logging. Click Components to for individual components, or click Groups to specify a log detail level components. Click a component or group name to select a log detail e cumulative; a level near the top of the list includes all the
Configuration Runtime	
General Properties	
🗖 Save runtime ch	anges to configuration as well
Change Log Detail Lev	els
Components	
Groups	*=info
	I ars com.ibm.i2ca *
	⊞ ≝_ com.ibm.port No Logging
	🕀 🚛 com.ibm.rela 🕮 Messages Only
	🕀 🕋 com.ibm.sec 🚢 All Messages and Traces
	Hat com.ibm.idask Message and Trace Levels
	E ≝_ com.ibm.wbi.*

\_\_\_\_h. You should see the below statement:

Configuration	Runtime		
General Pro	perties untime char Detail Levels	iges to configuration as well	
Grou	ponents IPS	*=info: com.ibm.j2ca.*=all	

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

# Part 4: Load trace file and review confidential messages

- \_\_\_\_\_1. Load log/trace file contents from new Server Logs view
  - \_\_\_\_a. Select Server Logs view located in the bottom pane

Construction Build Activities	Properties	🔝 Problems	🋅 Server Logs 🛛	위 Servers	谒 Asset Repositories

- \_\_\_\_b. From the icons menu under Server Logs, click 'Load Server Console or Log' icon ():
- \_\_\_\_ c. Ensure that 'Load from file' is selected and click Browse... next to Files
  - 1) In the 'Load From File' window, browse to **<WPS\_HOME>\logs\server1** and check the box for **trace.log**

🚯 Load From File	×
Load From File Select one or more log files. Click OK to load them in the Cross-Component Trace	
SystemErr.log	
Eiles:       C:\IBM\WID62\pf\wps\logs\server1\trace.log         Filter:       .log	

#### 2) Click OK

\_\_\_\_ d. You are back to 'Load Server Console or Log' window and the above selected trace file is populated here:

👍 Load Server Console or Log	
Load a server console or log files	
Select a server console or log files to load into the Server Logs view.	
Load from file	
Eiles: C:\QCert\WID62\pf\wps\logs\server1\trace.log <	Br <u>o</u> wse
C Load from server console	
Server: WebSphere Process Server v6.2 at localhost	~
C Load from server log <u>d</u> irectory	
Server: WebSphere Process Server v6.2 at localhost	~
Files:	Bro <u>w</u> se

#### \_\_\_\_e. Click Finish

2. Provide the input file:

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

- \_\_\_\_a. Copy the **sample.txt** file from <FFFILES> to **<EVENT\_DIR>/subdir**. The adapter will poll the copied file from the sub directory under event directory and will transfer it to the archive directory
- b. Because you have placed a file that contains customer business object, it will pass through the emitCustomerFile method and you should see this message in the console (or SystemOut.log):

SystemOut	Ο	********	***ENDPOINT emitCustomer*************
SystemOut	0	FILENAME :	SingleCustomerBO.xml
SystemOut	0	NAME>	IBM
SystemOut	0	ADDRESS>	11501 Burnet Rd
SystemOut	0	CITY>	Austin
SystemOut	0	STATE>	ТХ

\_\_\_\_ 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:\Labfiles62\LogAndTra	ce\arch	ivedir		<b>~</b>
Folders	×	Name 🔺	Size	Туре
LogAndTrace     Comparison     Comparison	-	SingleCustomerBO.xml.2008_08_19_16_04_52_062.success	1 KB	SUCCESS File

\_\_\_\_\_ d. The log/trace file should be loaded and displayed in the Server Logs view as shown below. The log records show all the messages of trace/log file. Note that some messages have 'FFRA001' in their description and this is because you specified the Adapter ID as 001 while configuring the adapter:

🚯 Build Activities 🔲 Proper	ties 🔝 Problems 👫 Servers	🔓 Asset Repos	sitories 🕒 C	ross-Component Trace 🛛 🗧	° 🗗
				<b>□</b> •   ‡ •   🗉   🖽 🕞   🞐   ↑ 🕂 🚭	S.
Trace : C:\IBM\\ser	ver1\trace.log				
Show All Log Records (Page 1	of 334)				
Туре	▲ Time	Thread ID	Description		
Log message	Oct 06, 2008 12:04:12.890	000000c5	ManagerAd	min I TRA50018I: The trace state has changed. The new trace state is *=info:com.ibm.j2ca	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.extension.eventmanagement.internal.EventManager pollForEvents(int) Ent	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.extension.eventmanagement.internal.SubscriptionManager recalculateSubs	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	< com.ibm.j2ca.extension.eventmanagement.internal.SubscriptionManager recalculateSubs	
🗏 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.extension.eventmanagement.internal.EventManager doRecovery Entering	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.extension.eventmanagement.internal.EventManager doTransactionsRecov	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	$< {\tt com.ibm.j2ca.extension.eventmanagement.internal.EventManager do Transactions Recov}$	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	< com.ibm.j2ca.extension.eventmanagement.internal.EventManager doRecovery Exiting m	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	logging	3 Start invoking EventPointImpl.getEventSource	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	3 com.ibm.j2ca.flatfile.FlatFileEventStoreWithXid getEvents Entering method getEvents of c	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.flatfile.FlatFileEventStoreWithXid getEvents eventStatus= 0	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	3 com.ibm.j2ca.flatfile.bridge.FlatFileBridge getEventsBasedOnSplitCriteria Entering method	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	3 com.ibm.j2ca.flatfile.util.FlatFileVerifier verifyEventDirectory Entering method verifyEvent	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	3 com.ibm.j2ca.flatfile.util.FlatFileVerifier verifyEventDirectory Exiting method verifyEventDi	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	logging	3 Start invoking EventPointImpl.getEventSource	
🗐 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	2 com.ibm.j2ca.extension.eventmanagement.internal.EventManager pollForEvents(int) No	
🕞 Log message	Oct 06, 2008 12:04:14.968	00000c3	FFRA001	< com.ibm.j2ca.extension.eventmanagement.internal.EventManager pollForEvents(int) Exi	

- \_\_\_\_3. Review trace file for confidential messages:
  - Flat File: When confidential tracing is enabled, BO data for both inbound and outbound is replaced by XXXXX. For outbound (create, append, and overwrite) the BO is not masked since the BO content is in the request and at the databinding, it is not possible to differentiate if confidential tracing was enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did not originate inside the EIS. This includes, but is not limited to: Transaction IDs, Event IDs, BO schemas, and Call sequences.
  - FTP: When confidential tracing is enabled: BO data for both inbound and outbound is replaced by XXXXX. Username is replaced by XXXXX. Password will always be written to the trace file as \*\*\*\*\* irrespective of whether confidential tracing is enabled or not. FTP URL is traced to the trace file. This is part of each trace message (for example: ftp://localhost/eventDir). This is required because when you are using multiple Activation Spec (AS) or MCFs with each AS polling one dir on the same FTP server, it is easier to identify to which AS the traces belong to. For outbound (create, append, and overwrite) the BO is not masked since the BO content is in the request and at the databinding it is not possible to differentiate if confidential tracing was enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did not originate inside the EIS. This includes, but is not limited to: Transaction IDs Event IDs BO Schemas Call sequences
  - EMail: When confidential tracing is enabled: BO data for both inbound and outbound is replaced by XXXXX. Username is replaced by XXXXX. Password will always be written to the trace file as \*\*\*\*\* irrespective of whether confidential tracing is enabled or not. Things that are not considered confidential for this purpose, includes anything that is internal to the adapter or runtime, and did not originate inside the EIS. This includes, but is not limited to: Transaction IDs Event IDs BO Schemas Call sequences
  - \_\_\_a. Open your windows explorer and open <WPS\_HOME>\logs\server1\trace.log using note pad
  - \_\_\_\_b. You should see 'XXXX' replaced for all the confidential data:

[10/3/08 11:58:04:015 CDT] 00000018 FFRA001	1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001	1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:58:04:015 CDT] 0000001 FFRA001	1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:58:04:015 CDT] 00000018 FRA001	com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:58:04:015 CDT] 00000018 FRA001 TraceActivationSpecProperties EP DataSource INDIN	1 com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:58:04:015 CDT] 00000018 FFRA001	< com.ibm.j2ca.flatfile.FlatFileResourceAdapter
[10/3/08 11:52:05:812 CDT] 000000e0 FFRA001	3 com.ibm.j2ca.flatfile.util.FlatFileUtil
[10/3/08 11:52:05:812 CDT] 000000e0 FFRA001 com.ibm.j2ca.flatfile.FlatFileEventStoreWithXi [10/3/08 11:52:05:812 CDT] 00000000 FFRA001	→ 3 d getRecordForEvent BO Content : ☎∞∞∞∞
com.ibm.j2ca.flatfile.FlatFileEventstorewithxi record object is C:\Labfiles62\LogAndTrace\eve	d getRecordForEvent DirectoryPath set in the

- 4. Restore the Sever Configuration
  - \_\_\_\_a. Right-click WebSphere Process Server v6.2 under the Servers view and select Add and remove projects... from the pop-up menu
  - \_\_\_\_b. Select LogAndTraceModuleApp under Configured projects and click < Remove
  - \_\_\_\_ c. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished

## What you did in this exercise

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

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

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

# 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.

Server     State     Status       Image: State     State     Synchronized	🗞 Build Activities 🔲 Properties 🔝 Problems 🔞 Ser	ver Logs 🦚 Servers 🛛 🕻 Asset	Repositories	🏇 🜔 🖉 🔳 🔑 🛅 🗖 🖗
📅 WebSphere Process Server v6.2 at localhost 🛱 Started Synchronized	Server 🔺	State	Status	
	🛃 WebSphere Process Server v6.2 at localhost	🛼 Started	Synchronized	
Terver	Ne <u>w</u>	· · · · · · · · · · · · · · · · · · ·	Server	
Open F3	Open	F3 T	VP	

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

🌐 New Server	
Define a New Server Choose the type of server to	o create
Server's <u>h</u> ost name:	<.rtp.raleigh.ibm.com
Select the server type:	Download additional server adapters
type filter text	
Runs service projects on the V	lication Server v6.0 lication Server v6.1 lication Server v7.0 Server v6.2 tal v6.0 Server tal v6.1 Server tal v6.1 Server on WAS 7 tess Server v6.2 WebSphere Process Server v6.2.
Server na <u>m</u> e:	WebSphere Process Server v6.2 at mvsxxx.rtp.raleigh.ibm.com
Server <u>r</u> untime environment:	WebSphere Process Server v6.2  Add
	Configure runtime environments
?	<back next=""> Einish Cancel</back>

\_\_\_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 Server Settings           Input settings for connecting to an existing WebSphere Application Server.					
WebSphere profile name:	<u></u>		Configure profiles		
Server connection types and administrative ports					
C Automatically determine connection settings					
Connection Type	Port	Default port	Description		
	Forc	2809	Designed to improve communication with the server		
SOAP	8880	8880	Designed to be more firewall compatible		
			1		
Run server with resources within the workspace					
✓ Security is enabled on this server					
Licer ID:		,,			
Bassword					
Pa <u>s</u> sworu.					
WebSphere server name:	server1				
Test Connection					
0		< [	ack Next > Einish Cancel		

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

🗞 Build Activities 🔲 Properties 🔝 Problems 🔞 Server Logs 🖗 Se	ervers 🛛 🕻 Asset Repositories	- 8
	参 🜔 🖉 🗉 🔡 🛅	
Server 🔺	State	Status
🔀 WebSphere Process Server v6.2 at localhost	🚡 Started	Synchronized
52 WebSphere Process Server v6.2 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.