

IBM WEBSPHERE ADAPTER 6.0.2 – LAB EXERCISE

## Custom Data Bindings Lab

What this exercise is about .....	2
Lab requirements .....	2
What you should be able to do .....	2
Introduction .....	3
Exercise instructions .....	3
Part 1: Preparation for the lab.....	4
1.1. Create database and subdirectory structure.....	5
1.2. Create module and import flat file RAR .....	7
1.3. Use WebSphere Process Server Administrative Console to configure data sources .....	11
1.4. Import custom bindings source code .....	16
1.5. Import and review business objects.....	17
Part 2: Inbound configuration - Delimited.....	21
2.1. Run ESD to create inbound interface and other artifacts .....	22
2.2. Create Java component with an interface .....	28
2.3. Create data and interface maps.....	31
2.4. Run intermediate test.....	39
Part 3: Outbound configuration - NameValue .....	41
3.1. Run ESD to create outbound interface and other artifacts .....	42
3.2. Create data and interface maps.....	47
3.3. Run Intermediate Test .....	56
Part 4: Outbound Configuration - FixedWidth.....	59
4.1. Run ESD to create Outbound Interface and other Artifacts.....	60
4.2. Create Data and Interface Maps.....	65
4.3. Run Intermediate Test .....	73
Part 5: Test the Scenario .....	76
Appendix: Troubleshooting Tips .....	79
What you did in this exercise .....	81

## What this exercise is about

The objective of this lab is to provide you with an understanding of how to configure Adapters for Custom Data Bindings. In this lab you will use three custom data bindings – Delimited, NameValue, and FixedWidth.

## Lab requirements

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

- WebSphere Integration Developer V6.0.2 installed
- WebSphere Process Server V6.0.2 Test Environment installed
- WebSphere Adapter for Flat File V6.0.2 installed
- Unzip LabFiles602.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:

- Configure and test FlatFile Adapter Inbound operation for Delimited custom bindings
    - Use ESD to configure FlatFile Adapter for inbound operation to process Delimited files
    - Create Java Component with an Interface
    - Create required Data maps and Interface map
    - Use WebSphere Integration Developer test environment to test the Delimited inbound scenario
  - Configure and test FlatFile Adapter Outbound operation for NameValue custom bindings
    - Use ESD to configure FlatFile Adapter for outbound operation to process NameValue files
    - Create required Data maps and Interface map
    - Use WebSphere Integration Developer test environment to test the NameValue outbound scenario
  - Configure and test FlatFile Adapter Outbound operation for FixedWidth custom bindings
    - Use ESD to configure FlatFile Adapter for outbound operation to process FixedWidth files
    - Create required Data maps and Interface map
    - Use WebSphere Integration Developer test environment to test the FixedWidth outbound scenario
  - Test FlatFile Adapter for both Inbound and Outbound with one Module
-

## Introduction

This lab provides the steps to test the FlatFile adapter for three different Custom Data Bindings – Delimited, NameValue and FixedWidth. This lab guides you to configure FlatFile Adapter to process Delimited files in its inbound operation and NameValue and FixedWidth files in its outbound operation.

## Exercise instructions

Some instructions in this lab may be 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 vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Reference Variable	Windows Location	AIX/UNIX Location
<WID_HOME>	C:\Program Files\IBM\ID\6.0	
<WPS_HOME>	<WID_HOME>\runtimes\bi_V6	
<FFADAPTER_HOME>	C:\Program Files\IBM\ResourceAdapters\FlatFiles\adapter\FlatFile	
<LAB_FILES>	C:\Labfiles602	
<WORKSPACE>	<LAB_FILES>\DTF\workspace	
<EVENT_DIR>	<LAB_FILES>\DTF\EventDir	
<ARCHIVE_DIR>	<LAB_FILES>\DTF\ArchiveDir	
<DATA_DIR>	<LAB_FILES>\DTF\DataDir	
<STAGING_DIR>	<LAB_FILES>\DTF\StagingDir	
<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, C:\LabFiles602\ would be replaced by C:/LabFiles602/

---

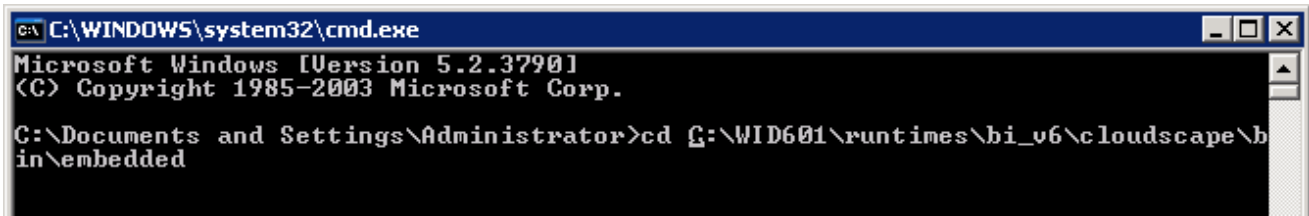
## Part 1: Preparation for the lab

In this part, you will create the required Database in cloudscape, data source using Administrative console and the directory structure. You will also import the FlatFile RAR file, source code, and the Business Objects into the project that will be created in this part.

## 1.1. Create database and subdirectory structure

This part of the lab, describes the steps for creating the **DTFDATABASE** database in Cloudscape which will contain the Event Persistence Table, **DTFTABLE**. The **DTFTABLE** will be created automatically for you during the install and deployment of the application containing the Flat File adapter to the WebSphere Process Server.

- \_\_\_ 1. Start the Cloudscape Cview Graphical User Interface (GUI) by executing the **cvview.bat** file
  - \_\_\_ a. Open a command prompt window, change to the following subdirectory  
<WPS\_HOME>\cloudscape\bin\embedded

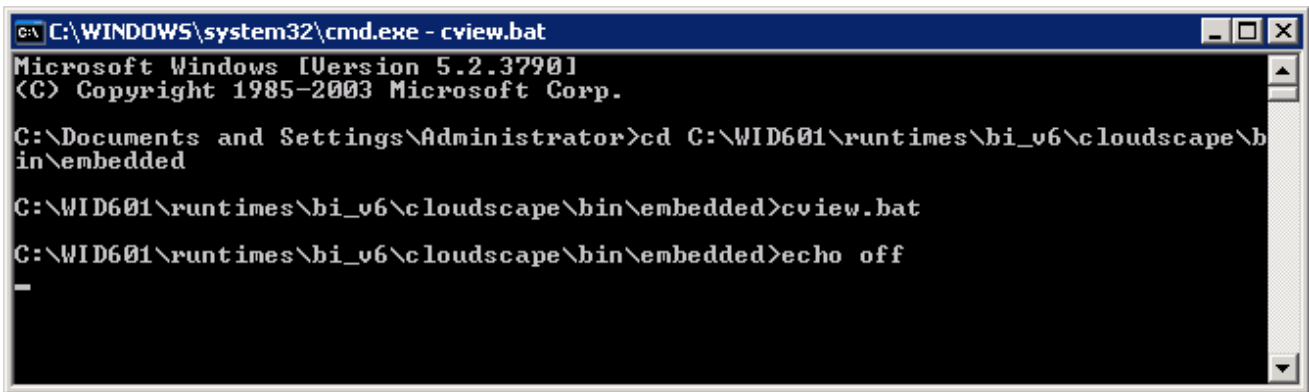


```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\WID601\runtimes\bi_v6\cloudscape\bin\embedded
  
```

- \_\_\_ b. Type **cvview.bat**



```

C:\WINDOWS\system32\cmd.exe - cvview.bat
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

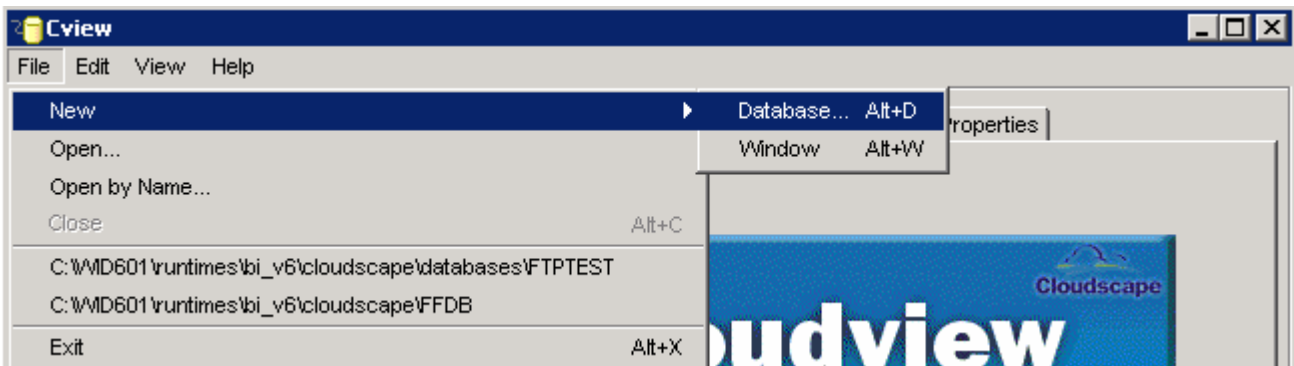
C:\Documents and Settings\Administrator>cd C:\WID601\runtimes\bi_v6\cloudscape\bin\embedded

C:\WID601\runtimes\bi_v6\cloudscape\bin\embedded>cvview.bat

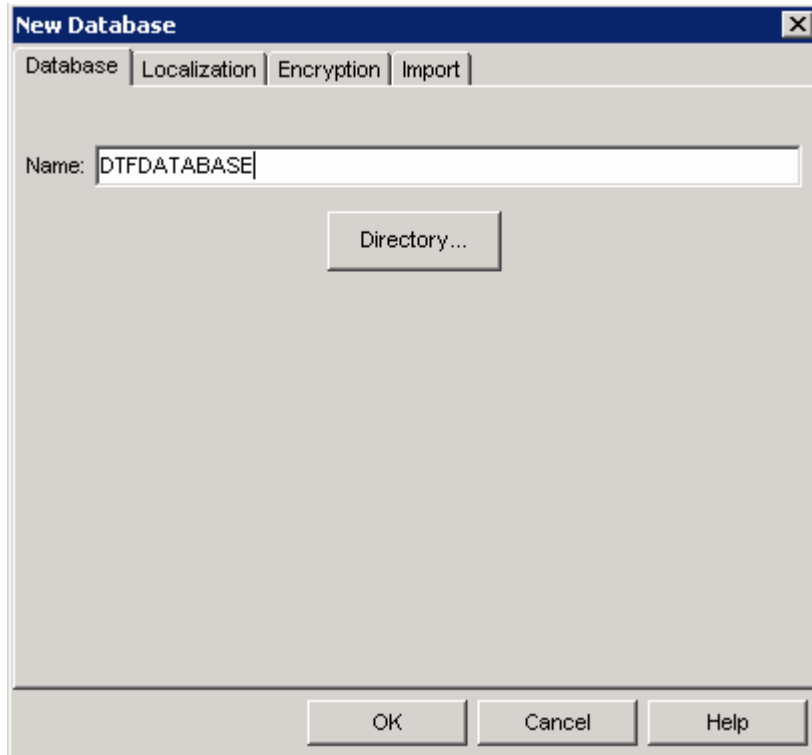
C:\WID601\runtimes\bi_v6\cloudscape\bin\embedded>echo off

-
  
```

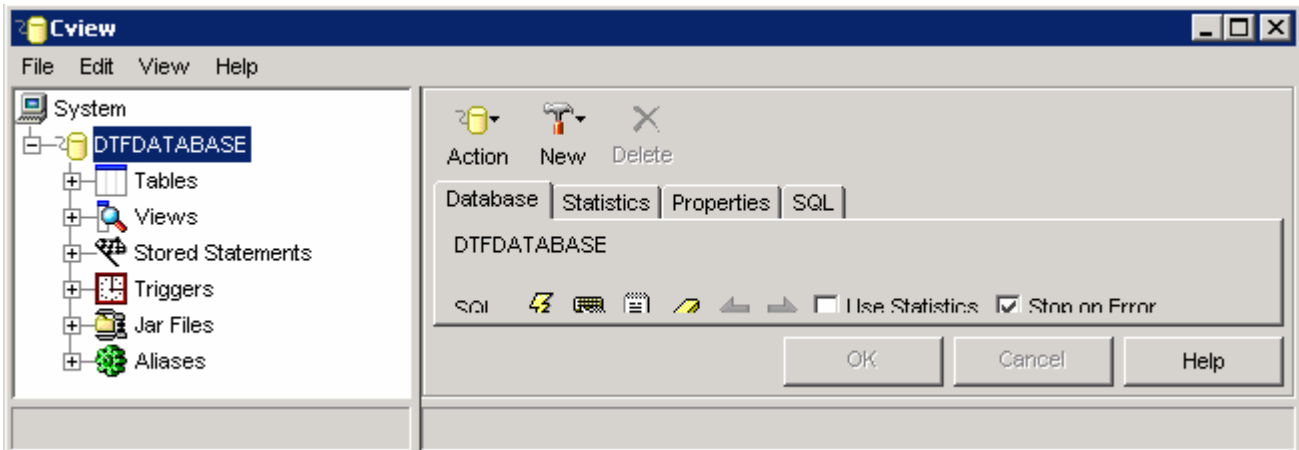
- \_\_\_ c. The Cview window will be opened. From that window, select **File > New > Database...**



\_\_ d. Enter **DTFDATABASE** in the Name field and click **OK**




You will see the **DTFDATABASE** created in the left pane of the Cview window:

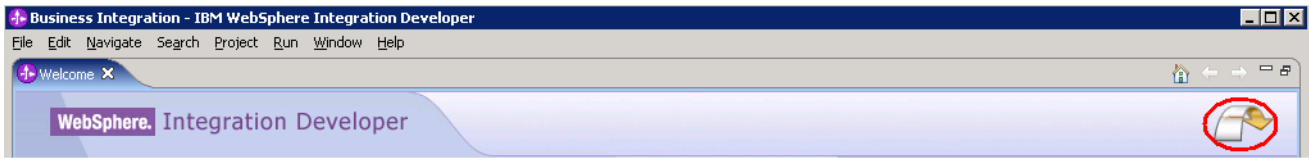


\_\_ e. Select **File > Exit** to close the Cview GUI

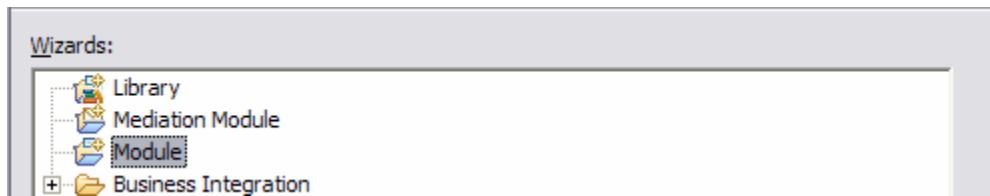
2. Unzip the provided LabFiles602.zip to your C:\ (root) drive, if you have not already done so. This will create the necessary subdirectory structure to complete the lab, and also provides you with sample text files.

## 1.2. Create module and import flat file RAR

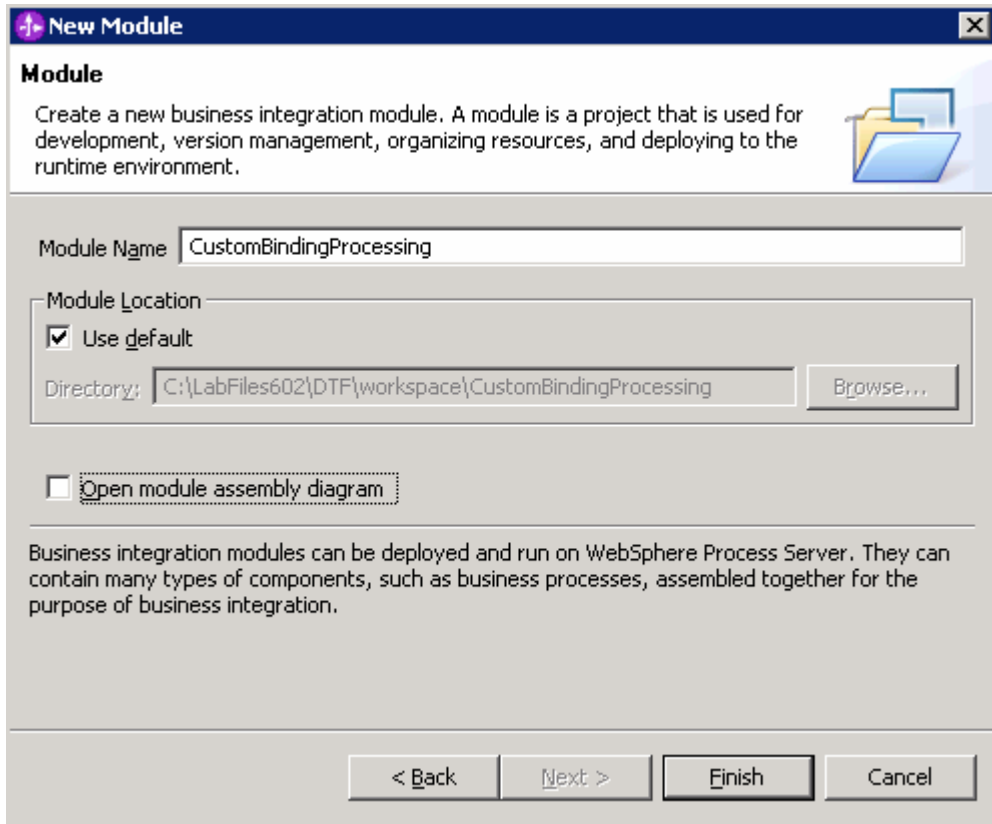
- \_\_\_ 1. Create CustomBindingProcessing module
  - \_\_\_ a. Start the WebSphere Integration Developer V6.0.2 with a new workspace if not already started.
    - 1) Select **Start > Programs > IBM WebSphere > Integration Developer V6.0.2 > WebSphere Integration Developer V6.0.2**
    - 2) From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field
    - 3) Click on the  button on the right hand corner to close the Welcome page and proceed with the workbench



- \_\_\_ b. From the main menu, select **File > New > Project**
- \_\_\_ c. From New Project window, select **Module** and click **Next**



- \_\_\_ d. Create New Module for the scenario
  - 1) From the New Module window, enter **CustomBindingProcessing** for the **Module Name**, uncheck **Open module assembly diagram**



2) Click **Finish**

You should see the CustomBindingProcessing module listed in the Business Integration perspective

\_\_\_ 2. Import FlatFile RAR into CustomBindingProcessing Module

\_\_\_ a. From the main menu, select **File > Import**

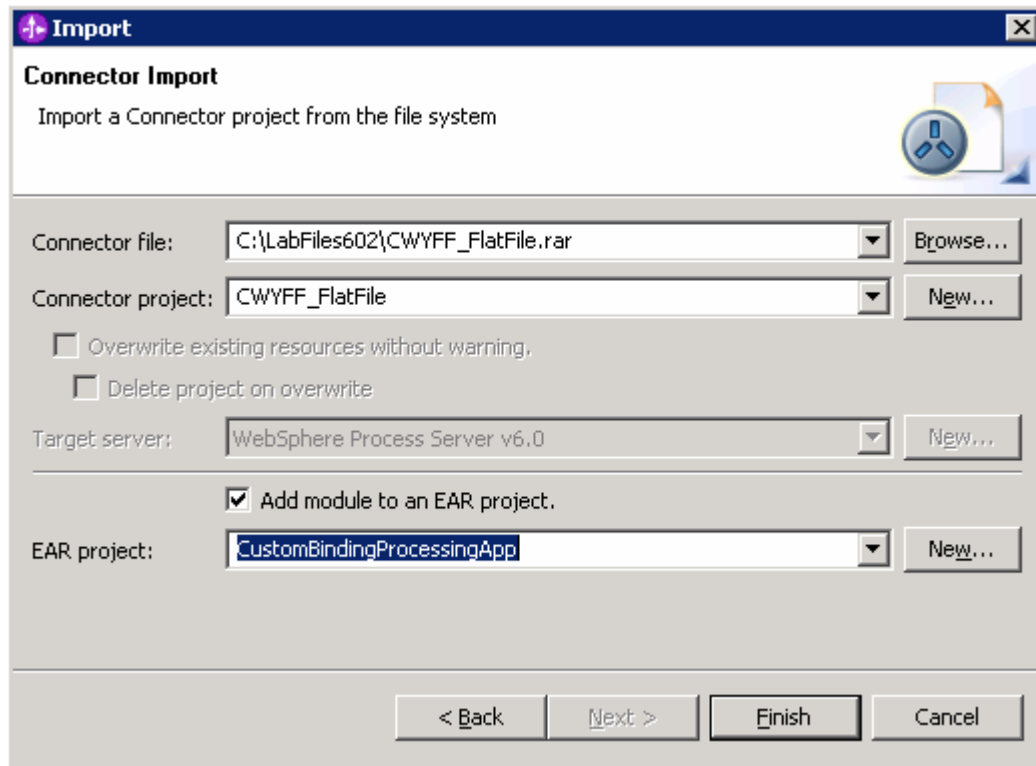
\_\_\_ b. From the Select screen, select **RAR File** and click **Next**

\_\_\_ c. Enter the following for the Connector Import screen

1) **Connector file:** Click on **Browse...** button and select **CWYFF\_FlatFile.rar** from **<FFADAPTER\_HOME>**

2) **EAR project:** Click on the drop down arrow and select **CustomBindingProcessingApp**





3) Click **Finish**

\_\_\_ d. Select **Yes** from Confirm Perspective Switch window to continue with the J2EE Perspective

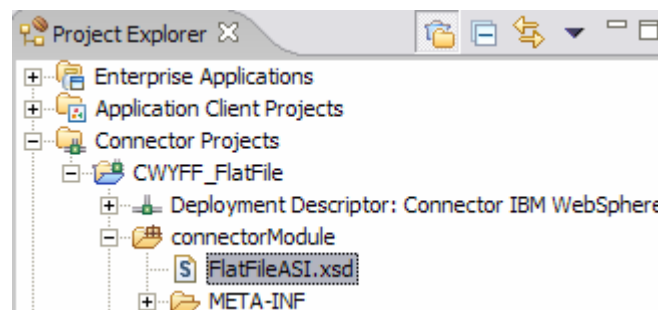
---

**Note:** You can also change to the J2EE perspective from the main menu: select **Window > Open Perspective > Other** and then select **J2EE** from the Select Perspective window.

---

\_\_\_ e. Modify FlatFileASI.xsd with the binding metadata information

1) From the Project Explorer, expand **Connector Projects > CWYFF\_FlatFile > connector Module**



2) Right-click on **FlatFileASI.xsd** and select **Open With > Text Editor** to open it in text editor

3) Add following highlighted piece of binding metadata information to FlatFileASI.xsd. This code can be added between **XMLBOSerializerDataBindingMetadata** and **DataTransformationMetadata** as shown.

**Note:** For your convenience, the metadata information to be added is provided in the file **BindingMetadata.txt** under **<LAB\_FILES>DTF**

```

<complexType name="XMLBOSerializerDataBindingMetadata">
<sequence minOccurs="1" maxOccurs="1">
  <element name="ContentType" type="string"
    fixed="text/xml" />
  <element name="DataBindingClass" type="string" fixed="com.ibm.j2ca.extension.emd.runtime.WBIXMLDataBinding"/>
</sequence>
</complexType>

<complexType name="Delimited1DataBindingMetadata">
<sequence minOccurs="1" maxOccurs="1">
  <element name="ContentType" type="string"
    fixed="text/delimited1" />
  <element name="DataBindingClass" type="string" fixed="com.ibm.websphere.samples.databinding.DelimitedEISDataBinding"/>
  <element name="TargetNamespace" type="string"/>
  <element name="Delimiter" type="string"/>
</sequence>
</complexType>

<complexType name="FixedWidth1DataBindingMetadata">
<sequence minOccurs="1" maxOccurs="1">
  <element name="ContentType" type="string"
    fixed="text/fixedwidth" />
  <element name="DataBindingClass" type="string" fixed="com.ibm.websphere.samples.databinding.FixedWidthEISDataBinding"/>
  <element name="TargetNamespace" type="string"/>
  <element name="FieldWidth" type="string"/>
  <element name="PadCharacter" type="string"/>
</sequence>
</complexType>

<complexType name="NameValueDataBindingMetadata">
<sequence minOccurs="1" maxOccurs="1">
  <element name="ContentType" type="string"
    fixed="text/namevalue" />
  <element name="DataBindingClass" type="string" fixed="com.ibm.websphere.samples.databinding.NameValueEISDataBinding"/>
  <element name="TargetNamespace" type="string"/>
</sequence>
</complexType>

<complexType name="DataTransformationMetadata">
<sequence minOccurs="1" maxOccurs="1">

```

\_\_\_ f. Save (**Ctrl + S**) your changes to FlatFileAS1.xsd and close it by clicking on the X

\_\_\_ g. Change to Business Integration Perspective

- 1) Select **Window > Open Perspective > Other...**
- 2) Select Business Integration and click **OK**

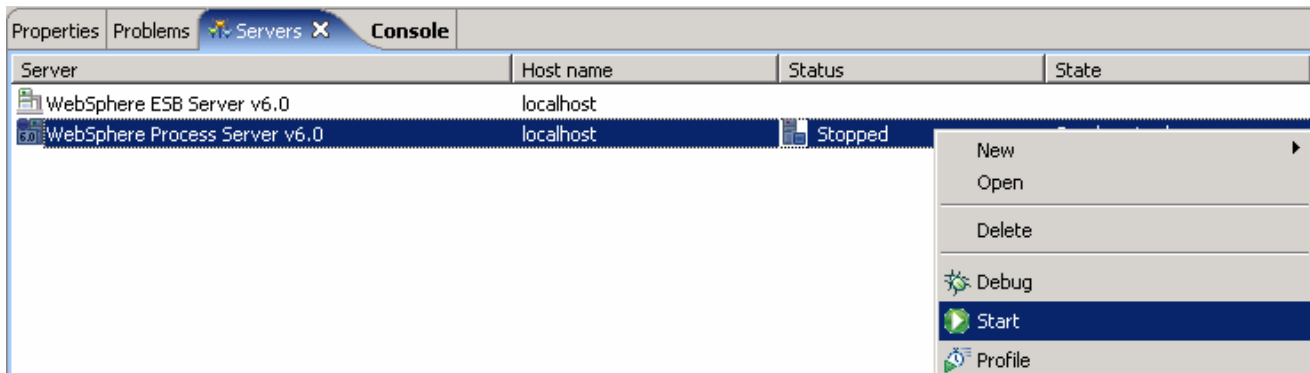
### 1.3. Use WebSphere Process Server Administrative Console to configure data sources

In this part of the lab, you will make use of WebSphere Process Server Administrative Console to create the required JDBC Provider and the Data source that will be used by the Adapter to configure itself to the end point.

1. Start the WebSphere Process Server

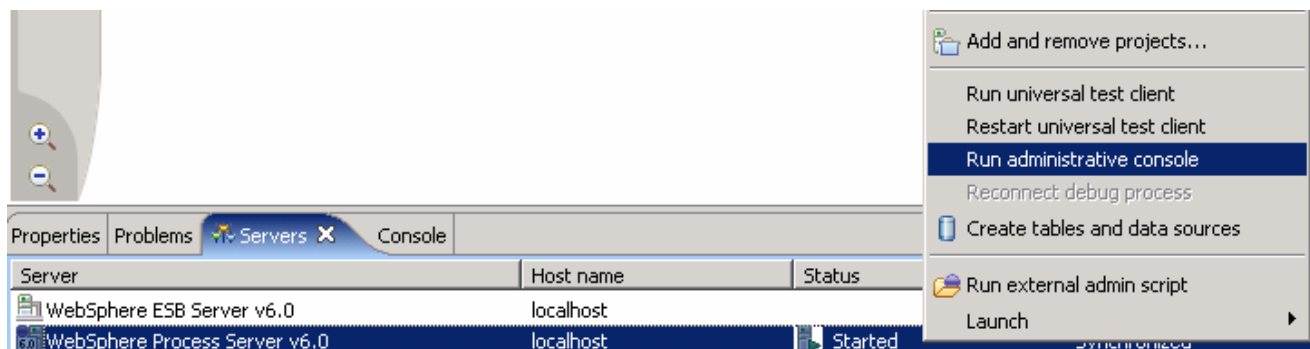
a. Select the **Servers** view

b. Right-click on the row that contains **WebSphere Process Server V6.0** and select **Start** from the context menu



c. Wait until the server status shows as **Started**

2. Right-click on the row that contains WebSphere Process Server V6.0 and select **Run administrative console** from the context menu

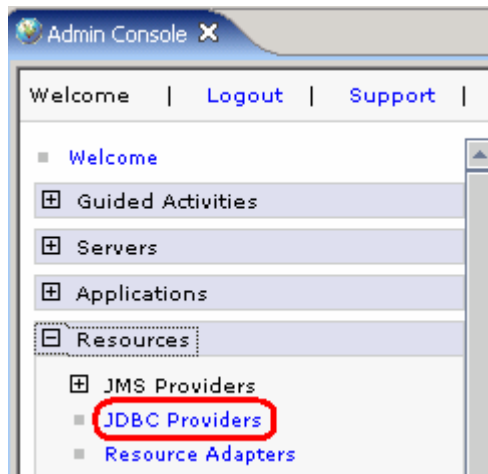


3. From the Administrative Console, enter any **User ID** and click on **Log In**

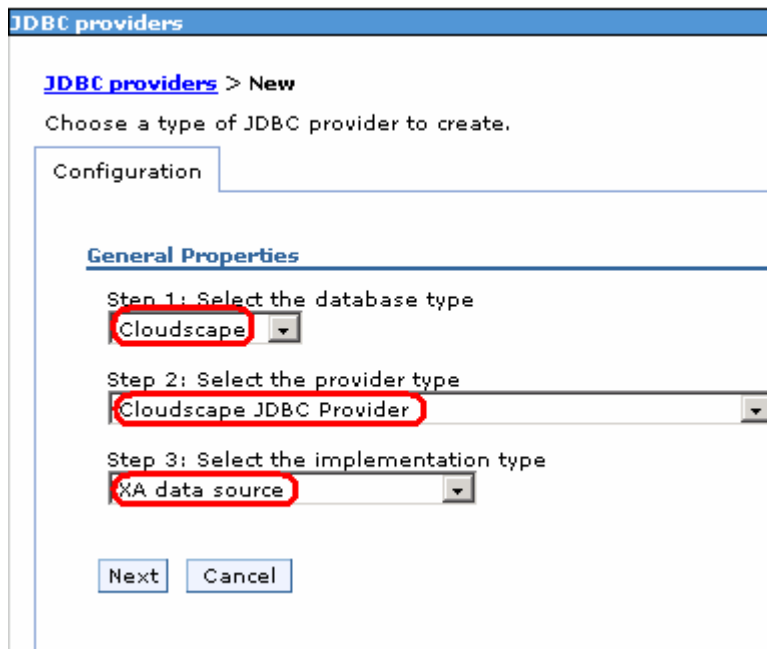
Welcome, please enter your information.

User ID:

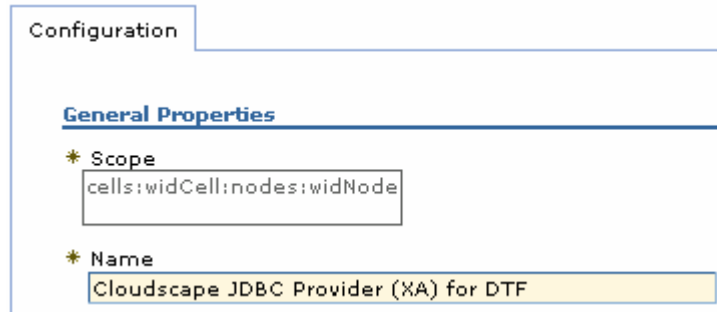
- \_\_\_ 4. Expand **Resources** on the left pane and select **JDBC Providers**



- \_\_\_ 5. Accept the default scope and click on **New**
- \_\_\_ 6. In the following screen, for General Properties, select the following from the dropdown list:
- \_\_\_ a. Step 1: **Cloudscape**
  - \_\_\_ b. Step 2: **Cloudscape JDBC Provider**
  - \_\_\_ c. Step 3: **XA data source**
- \_\_\_ 7. Click **Next**



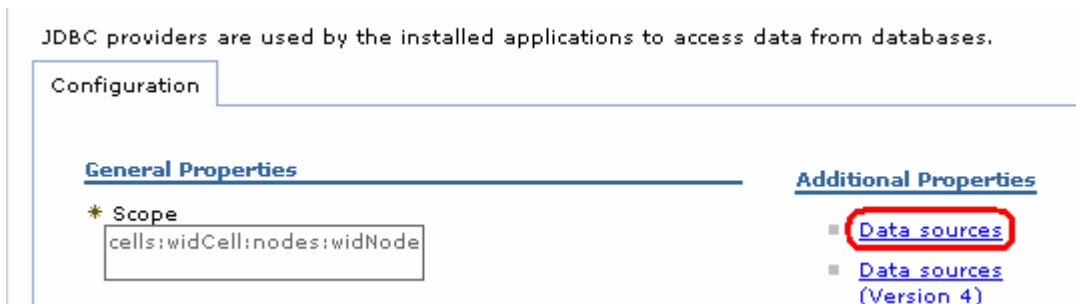
\_\_\_ 8. Enter **Cloudscape JDBC Provider (XA) for DTF** in the **Name** field and then click **OK**



\_\_\_ 9. Click **Save** on the top of the window, and then click on **Save** from the following screen

\_\_\_ 10. Click **Cloudscape JDBC Provider (XA) for DTF** from the following screen to create a new data source

\_\_\_ 11. Select **Data sources** under **Additional Properties** on the right hand side



\_\_\_ 12. Create the required JNDI Data source

\_\_\_ a. Click **New**

\_\_\_ b. Enter these values:

a) Name: **Cloudscape JDBC Driver XA DataSource for DTF**

b) JNDI Name: **jdbc/DTF**

c) Database name: **DTFDATABASE**

\_\_ c. Click **OK**

**\* Name**  
Cloudscape JDBC Driver XA DataSource for DTF

**JNDI name**  
jdbc/DTF

Use this Data Source in container managed persistence (CMP)

**Description**  
New JDBC Datasource

**Category**

**Data store helper class name**

Select a data store helper class  
Data store helper classes provided by WebSphere Application Server

Cloudscape data store helper  
(com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper)

Specify a user-defined data store helper  
Enter a package-qualified data store helper class name

**Component-managed authentication alias**  
Component-managed authentication alias  
(none)

**Authentication alias for XA recovery**

Use component-managed authentication alias

Specify:  
(none)

**Container-managed authentication**  
Container-managed authentication alias (deprecated in V6.0, use resource reference authentication settings instead)  
(none)

Mapping-configuration alias (deprecated in V6.0, use resource reference authentication settings instead)  
(none)

**Cloudscape data source properties**

**\* Database name**  
DTFDATABASE

- \_\_\_ 13. Click on **Save** and then **Save** from the following screens
- \_\_\_ 14. Test the Data Source connection
  - \_\_\_ a. Check the box next to **Cloudscape JDBC Driver XA DataSource for DTF** and click on **Test connection** from the top of the screen

[JDBC providers](#) > [Cloudscape JDBC Provider \(XA\) for DTF](#) > **Data sources**

A data source is used by the application to access data from the database. A data source is created under a JDBC provider, which supplies the specific JDBC driver implementation class.

☏ Preferences

New	Delete	<b>Test connection</b>	Manage state...	
<input checked="" type="checkbox"/>	Cloudscape JDBC Driver XA DataSource for DTF	jdbc/DTF	New JDBC Datasource	
Total 1				

- \_\_\_ b. You should see the following success message on the top of the screen:

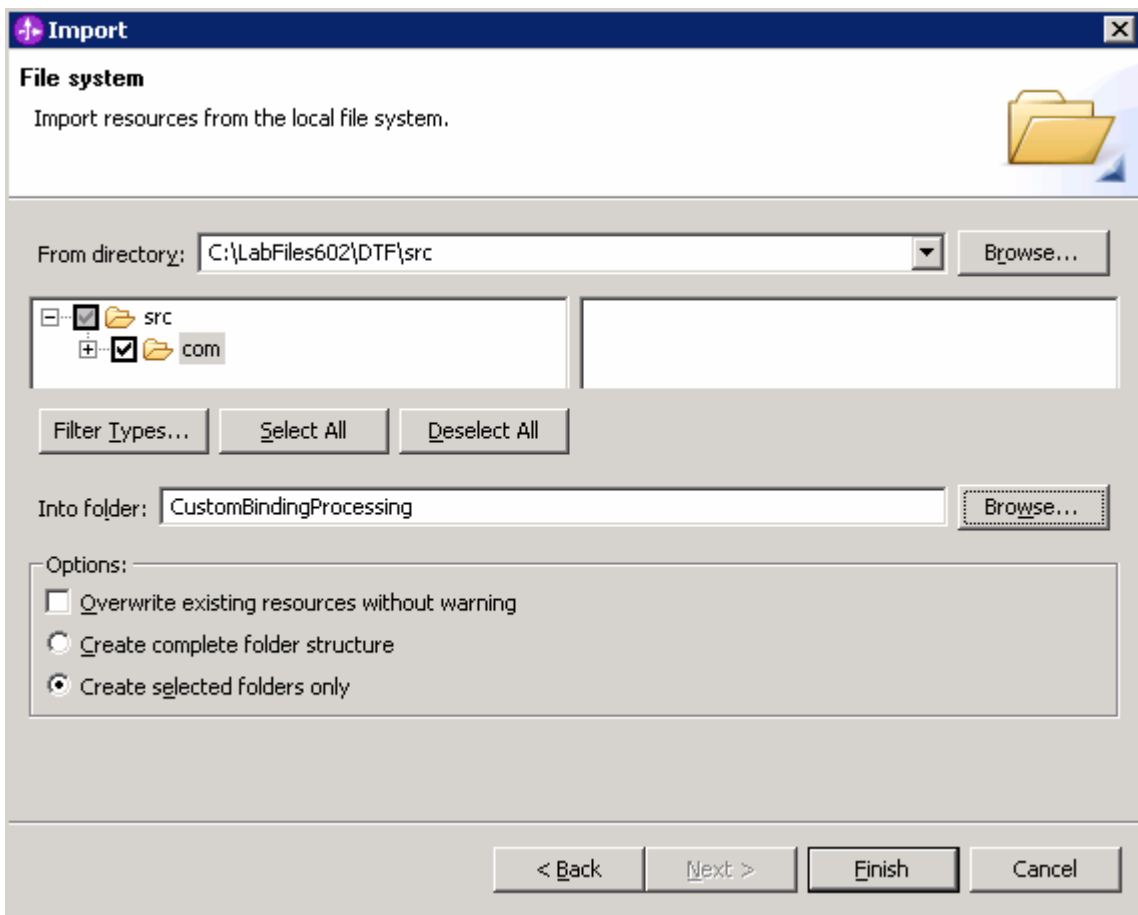
☏ Messages

**i** Test connection for data source Cloudscape JDBC Driver XA DataSource for DTF on server server1 at node widNode was successful.

- \_\_\_ 15. Log out of the Administrative Console and close it

## 1.4. Import custom bindings source code

- \_\_\_ 1. From main menu, select **File > Import**
- \_\_\_ 2. select **File System** from the list in the Select screen and click **Next**
- \_\_\_ 3. Enter the following for the File system screen
  - \_\_\_ a. **From Directory:** Click on **Browse** and select <LAB\_FILES>\DTF\src.
  - \_\_\_ b. Click **OK**
  - \_\_\_ c. Expand **src** and add a checkmark next to **com** as shown.
  - \_\_\_ d. **Into Folder:** Click on **Browse** and an Import Into Folder window will be opened
  - \_\_\_ e. Select **CustomBindingProcessing** and click **OK**
  - \_\_\_ f. Click **Finish** from the Import window



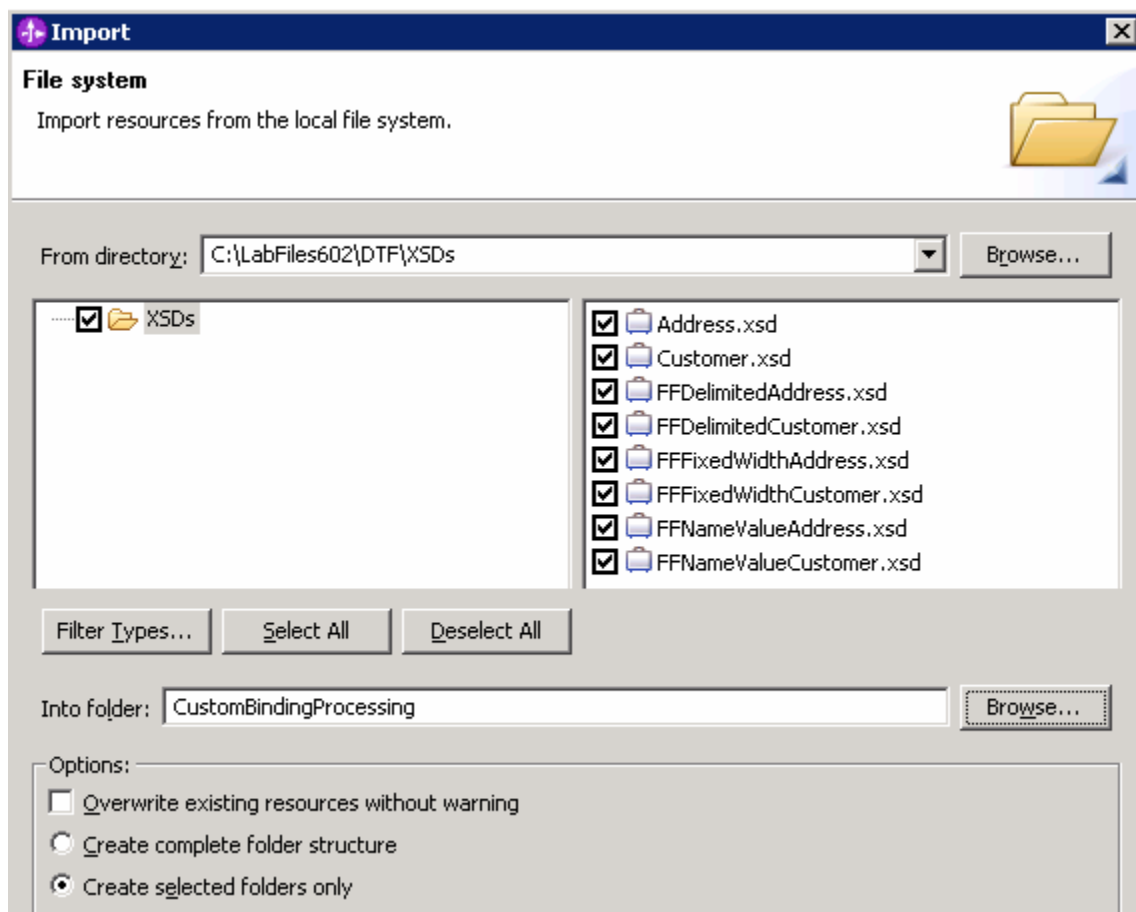
**Note:** Ignore the errors that you will see after this step. These errors will be resolved in the following steps as you continue with the lab.



## 1.5. Import and review business objects

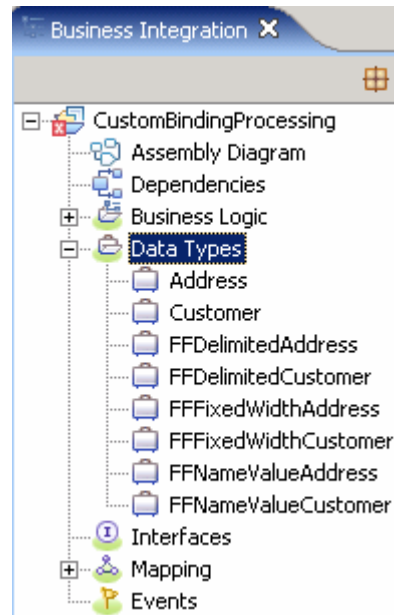
In this part you will import the required business objects – Address, Customer, FFDelimitedAddress, FFDelimitedCustomer, FFNameValueAddress, FFNameValueCustomer, FFFixedWidthAddress, and FFFixedWidthCustomer and then review the attributes of each of those Business Objects.

- \_\_\_ 1. From main menu, select **File > Import**
- \_\_\_ 2. select **File System** from the list in the Select screen and click **Next**
- \_\_\_ 3. Enter the following for the File system screen
  - \_\_\_ a. **From Directory:** Click on **Browse** and select **<LAB\_FILES>DTF\XSDs**. Click **OK**.
  - \_\_\_ b. Click on **Select All** to select all the .xsd files on the right side window
  - \_\_\_ c. **Into folder:** Click on **Browse** and an Import Into Folder window will be opened
  - \_\_\_ d. Select **CustomBindingProcessing** and click **OK**
- \_\_\_ 4. Click **Finish** from the Import window



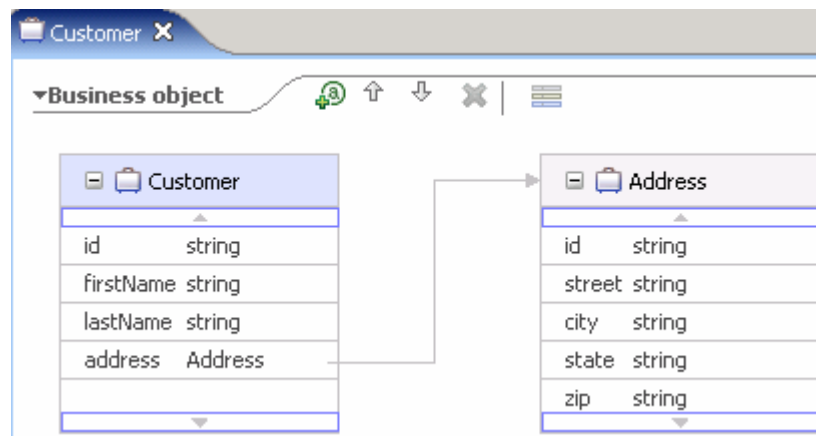
- \_\_\_ 5. Review the imported Business objects
  - \_\_\_ a. From the Business Integration view, expand **CustomBindingProcessing > Data Types** and observe the imported Data Types

IBM WebSphere Adapter 6.0.2 – Lab Exercise



\_\_ b. Review Customer and Address BOs

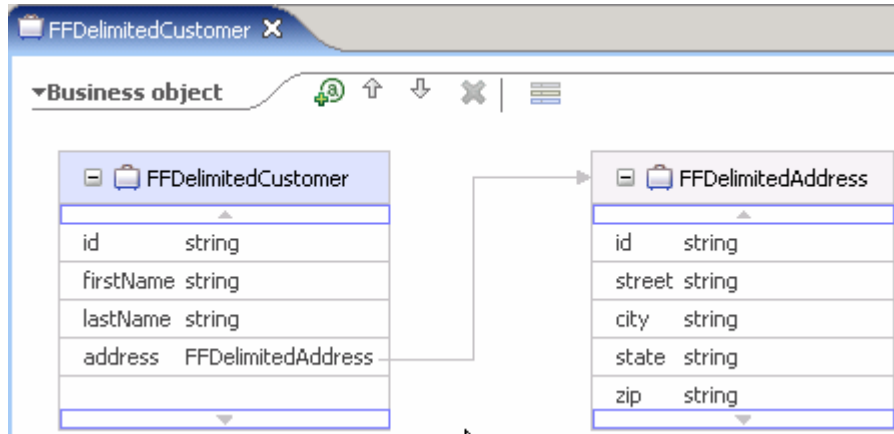
- 1) Double-click on **Customer** to open it in the Business object editor on the right side. Expand **Address** from the Customer BO editor window on the right side.



- 2) Click on **X** next to Customer to close it

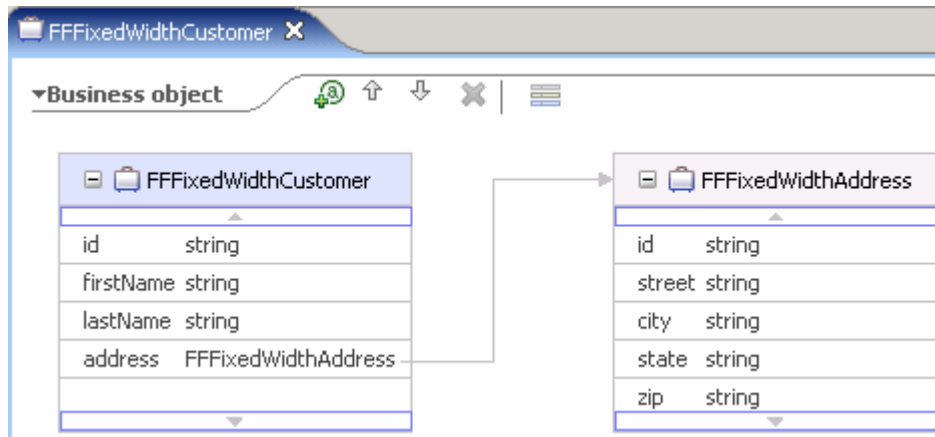
\_\_ c. Review FFDelimitedCustomer and FFDelimitedAddress BOs

- 1) Double-click on **FFDelimitedCustomer** to open it in the Business object editor on the right side. Expand **FFDelimitedAddress** from the FFDelimitedCustomer BO editor window on the right side.
- 2) Click on **X** next to FFDelimitedCustomer to close it



\_\_ d. Review FFFixedWidthCustomer and FFFixedWidthAddress BOs

- 1) Double-click on **FFFixedWidthCustomer** to open it in the Business object editor on the right side. Expand **FFFixedWidthAddress** from the FFFixedWidthCustomer BO editor window on the right side.

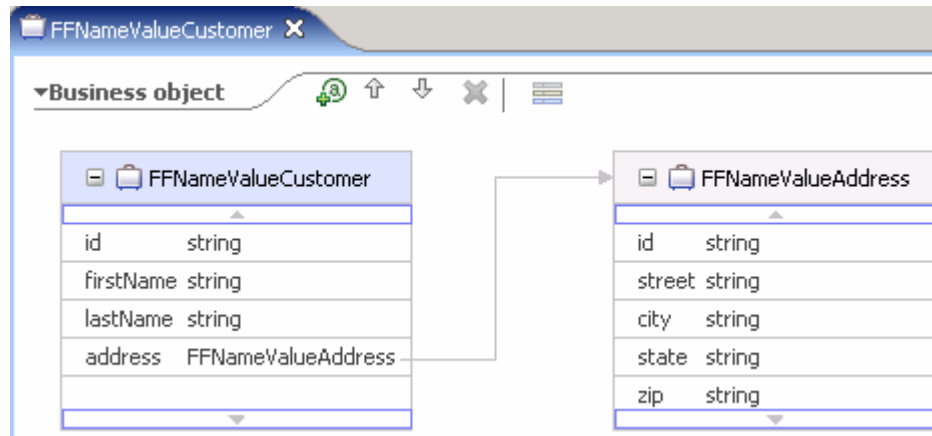


- 2) Click on **X** next to FFFixedWidthCustomer to close it

\_\_ e. Review FFNameValueCustomer and FFNameValueAddress BOs

- 1) Double-click on **FFNameValueCustomer** to open it in the Business object editor on the right side. Expand **FFNameValueAddress** from the FFNameValueCustomer BO editor window on the right side.

IBM WebSphere Adapter 6.0.2 – Lab Exercise



2) Click on X next to FFNameValueCustomer to close it

---

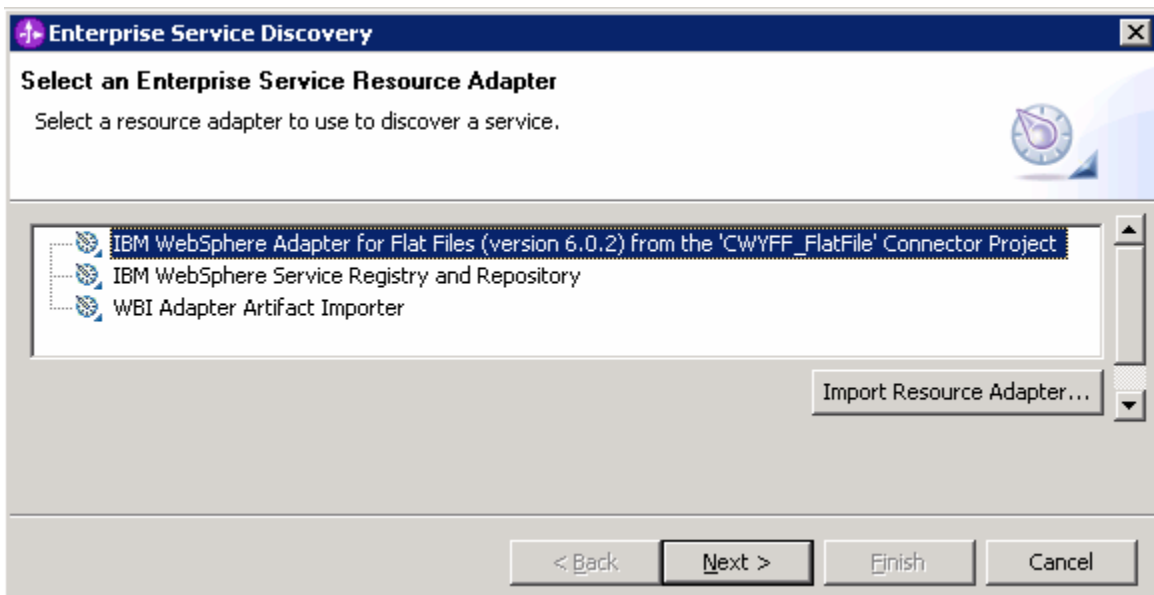
## Part 2: Inbound configuration - Delimited

In this part, you will perform these main tasks:

- You will use Enterprise Service Discovery (ESD) wizard to generate the FlatFileDelimitedInboundInterface, and also the artifacts – FFDelimitedCustomerWrapper, FFDelimitedCustomerWrapperBG and so forth.
- You will create an Interface, CustomerInterface, which you will add to the Java component, ProcessComponent.
- You will create these maps:
  - Data maps – FFDelimitedAddressToAddress, FFDelimitedCustomerToCustomer, and FFDelimitedCustomerWrapperBGToCustomer
  - Interface Map – DelimitedCustomerToCustomer
- You will test the adapter for the Delimited Inbound scenario

## 2.1. Run ESD to create inbound interface and other artifacts

- \_\_\_ 1. Launch the Enterprise Service Discovery wizard
  - \_\_\_ a. From the top Menu bar, select **File > New > Enterprise Service Discovery**
- \_\_\_ 2. Select **IBM WebSphere Adapter for Flat Files (version 6.0.2)** from the 'CWYFF\_FlatFile' Connector Project and click **Next**



- \_\_\_ 3. Enter the following for the Configure Settings for Discovery Agent screen
  - \_\_\_ a. **Folder Name:** Click on Browse and select **<WORKSPACE>\CustomBindingProcessing** and then click **OK**.
  - \_\_\_ b. For **Character set**, select **UTF-8** from the drop down menu.
  - \_\_\_ c. **Content Type:** Select **text/delimited1** from the drop down menu and the **DataBinding Type** will automatically be selected as the **Delimited1DataBinding**
  - \_\_\_ d. Enter the following for the **DataBindingProperties**
    - 1) TargetNameSpace: **http://www.ibm.com/crm**
    - 2) Delimiter: **,**
  - \_\_\_ e. Click on **Show Advanced** to see the logging properties. Note the **Log file output location**.

**Enterprise Service Discovery**

**Configure Settings for Discovery Agent**

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

Connection Configuration

Folder Name: C:\LabFiles602\DTF\workspace\CustomBindingProcessing

Character Set: UTF-8

Content Type: text/delimited1

DataBinding type: Delimited1DataBinding

Specify BO Properties

BiDi Properties

BiDi transformation

BiDi ordering schema: Implicit

Text direction: LTR

BiDi Symmetric Swapping

BiDi shapping: Nominal

BiDi numeric shapping: Nominal

DataBinding properties

TargetNamespace: http://www.ibm.com/crm

Delimiter: ,

DataBindingClass: com.ibm.websphere.samples.databinding.DelimitedEISDataBinding

Logging options

Log file output location:\* C:\LabFiles602\DTF\workspace\.metadata\FlatFileMetadataDiscoveryImpl.log

Logging Level: SEVERE

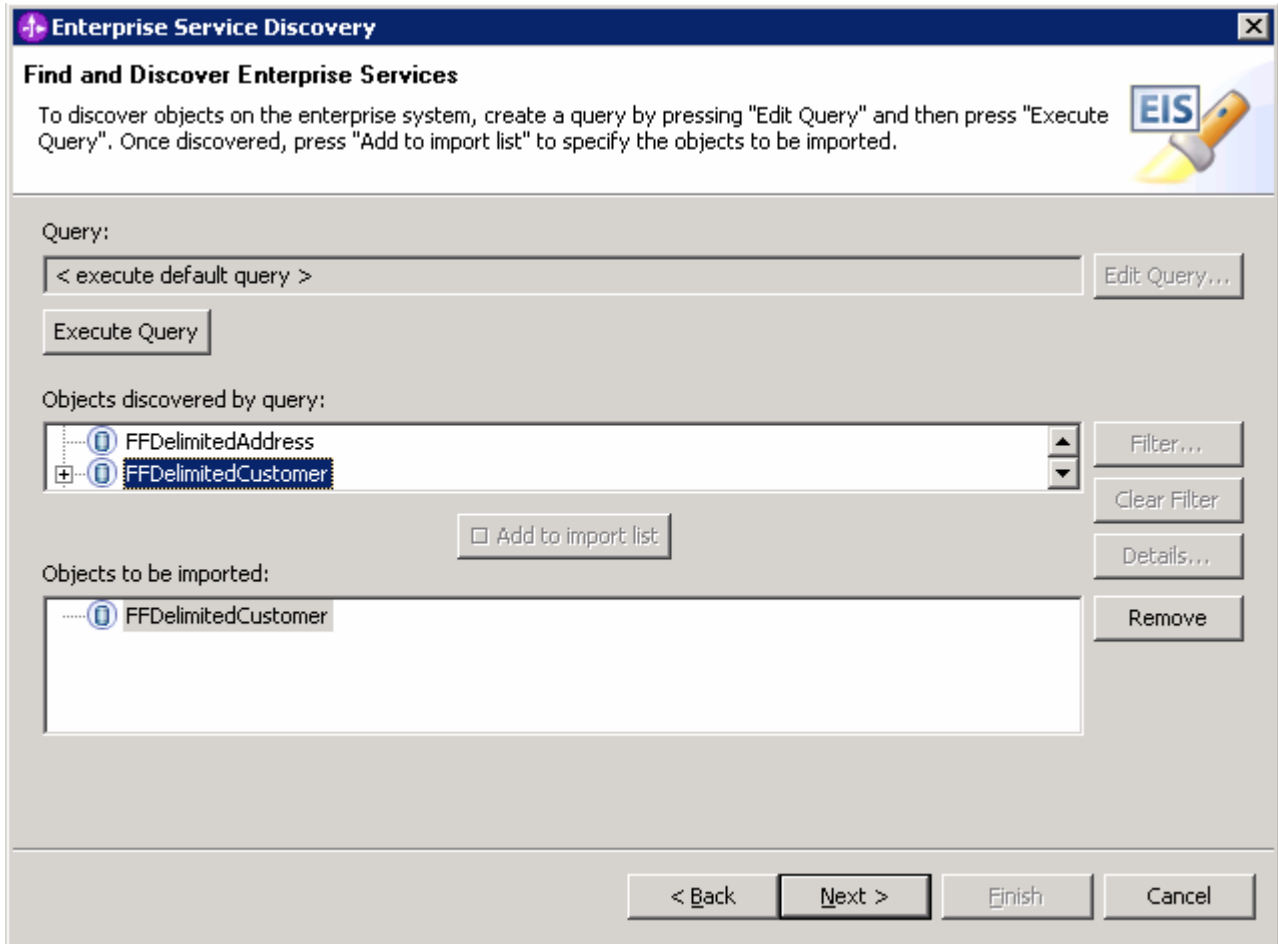
< Back

\_\_\_ f. Click **Next**

\_\_\_ 4. Find and discover the enterprise services. In this step you will select the business objects and services to be used with the adapter

\_\_\_ a. From Find and Discover Enterprise Services window, click on **Execute Query** button. You will see all the business objects the you imported under **Objects discovered by query**

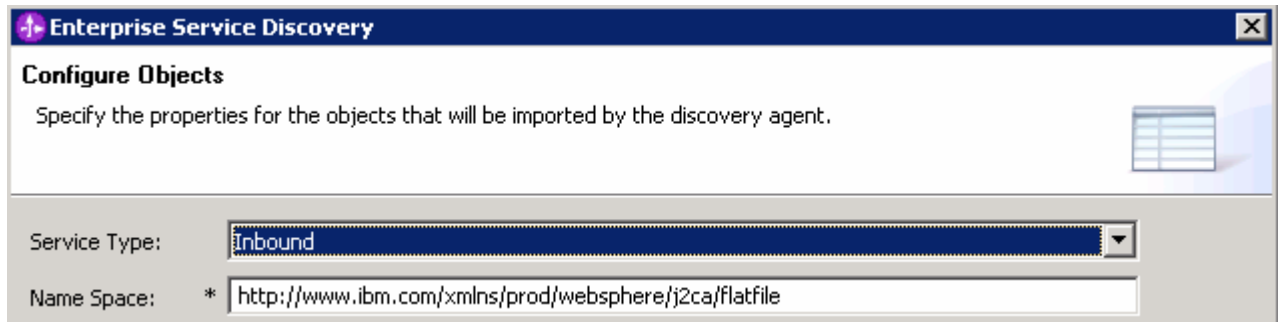
- \_\_\_ b. Select **FFDelimitedCustomer** business object and click **>>Add to import list** button. The Customer business object will now be displayed under the **Objects to be imported**



- \_\_\_ c. Click **Next**

\_\_\_ 5. Configure the objects that will be imported by the discovery agent

- \_\_\_ a. From the Configure objects window, for the Service Type, select **Inbound** from the dropdown menu.



- \_\_\_ b. Click **Next**

\_\_\_ 6. Enter the following on the Generate Artifacts window



\_\_ a. Enter the following for **Properties for Interface**

- 1) For **Module**, ensure that **CustomBindingProcessing** is selected.
- 2) **Name**: Enter **FlatFileDelimitedInboundInterface**

Properties for service

Module: CustomBindingProcessing [New...]

Namespace: http://CustomBindingProcessing/FlatFileDelimitedInboundI

Use default namespace

Folder: [Browse...]

Name: \* FlatFileDelimitedInboundInterface

Description: [ ]

\_\_ b. Ensure that **Use discovered connection properties** radio button is selected

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

Use connection properties specified on server

Use discovered connection properties

\_\_ c. Enter the following for **FlatFile Properties**

- 1) Event Directory: **<EVENT\_DIR>**
- 2) Archive Directory: **<ARCHIVE\_DIR>**
- 3) DataSource JNDI Name: **jdbc/DTF**
- 4) Event Table Name: **CustomDataBinding**
- 5) Event Content Type: **text/delimited1**
- 6) Default Object Name: **FFDelimitedCustomerWrapperBG**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

FlatFile Properties

Event Directory:	*	C:\LabFiles602\DTF\EventDir
<input checked="" type="checkbox"/> Archiving Processed		
Archive Directory:	*	C:\LabFiles602\DTF\ArchiveDir
Failed Archive Extension:		fail
Original Archive Extension:		original
Success Archive Extension:		success
Event File Mask:	*	*.*
<input type="checkbox"/> File Pass By Reference		
<input type="checkbox"/> Include End BO Delimiter		
<input checked="" type="checkbox"/> Create Table		
DataSource JNDI Name:	*	jdbc/DTF
Event Table Name:	*	CustomDataBinding
Database Schema Name:		
Database User Name:		
Database Password:		
Sort Event Files:		
Split Criteria:		
Splitting Function Class Name:		
Event Content Type:		text/delimited1
File Content Encoding:		
Default Object Name:		FFDelimitedCustomerWrapperBG

\_\_\_ d. Enter the following for **ResourceAdapterProperties**

- 1) Log file name: **C:\FFRA\DTF\Inlog.txt**
- 2) Trace file name: **C:\FFRA\DTF\Intrace.txt**

## IBM WebSphere Adapter 6.0.2 – Lab Exercise

ResourceAdapter Properties	
Logging and Tracing	
Adapter ID [String]:	* ResourceAdapter
Log file size [Integer]:	0
Log file name [String]:	C:\FFRA\DTF\Inlog.txt
Log Files [Integer]:	1
Trace file size [Integer]:	0
Trace file name [String]:	C:\FFRA\DTF\Intrace.txt
Trace files [Integer]:	1

- \_\_ e. Click **Finish**
- \_\_ f. From Business Integration window, expand **CustomBindingProcessing > Data Types**
- \_\_ g. Right-click on **FFDelimitedCustomer** and select **Open With > Text Editor** to open it
- \_\_ h. Delete the following line highlighted in blue with include statement:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xi
  <xsd:include schemaLocation="FFDelimitedAddress.xsd" />
  <include schemaLocation="FFDelimitedAddress.xsd" />
```

- \_\_ i. Save (**Ctrl+S**) your changes and close FFDelimitedCustomer BO

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

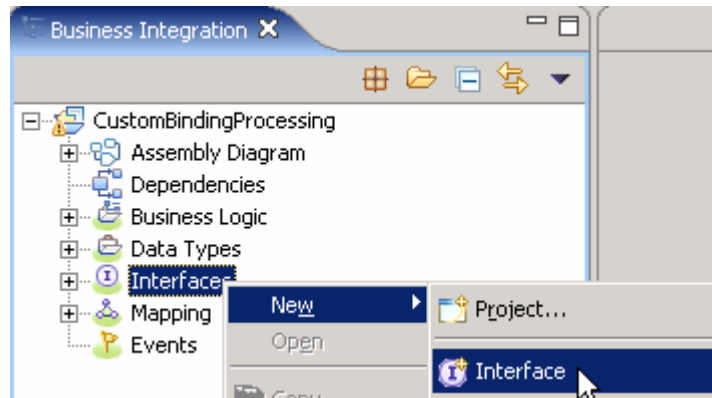
---

## 2.2. Create Java component with an interface

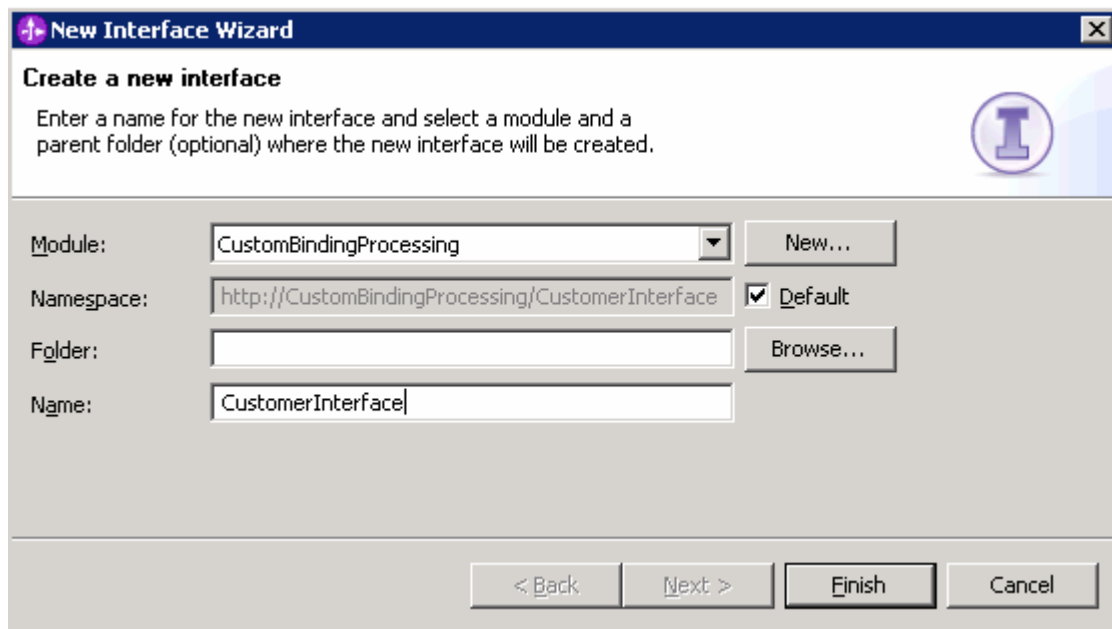
### 1. Create Interface

a. From Business Integration window, expand **CustomBindingProcessing > Interfaces**


b. Right-click on **Interfaces** and select **New > Interface**



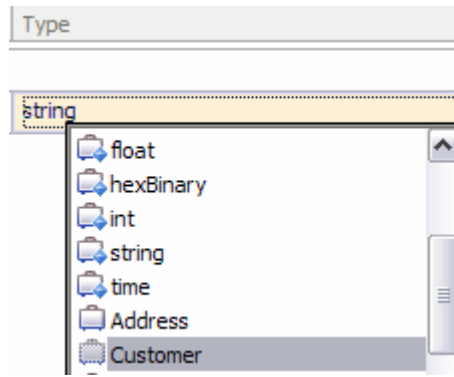
c. From New Interface Wizard, for **Name** enter **CustomerInterface** and click **Finish**



d. Add **processCustomer** one way operation


- 1) Click on **Add one way operation** (  ) icon
- 2) Replace operation1 with **processCustomer**
- 3) Replace input1 with **customer** for Name
- 4) Click on **string** under Type and select **Customer** from the context menu

IBM WebSphere Adapter 6.0.2 – Lab Exercise









\_\_ e. Repeat step d to add **appendCustomer** one way operation








\_\_ f. Add Request Response Operation, **existsCustomer**

- 1) Click on **Add Request Response Operation** (  )icon
- 2) Replace operation1 with **existsCustomer**
- 3) Replace input1 with **customer** for Name
- 4) Click on **string** under Type and select **Customer** from the context menu
- 5) Replace output1 with **response** for Name
- 6) Leave the Type as **string**

You should see the following screen after you are done with creating all the three operations

▼ **Operations**   |    

Operations and their parameters

	Name	Type
▼ 	processCustomer	
	customer	Customer
▼ 	appendCustomer	
	customer	Customer
▼ 	existsCustomer	
	customer	Customer
	response	string

\_\_ g. Save (**Ctrl+S**) your changes and close CustomerInterface

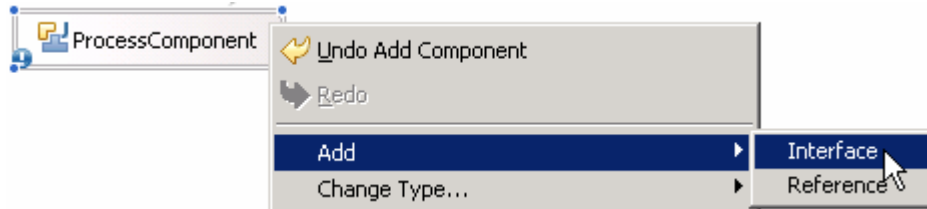
\_\_\_ 2. Create Java Component

\_\_ a. Open Assembly diagram for CustomBindingProcessing

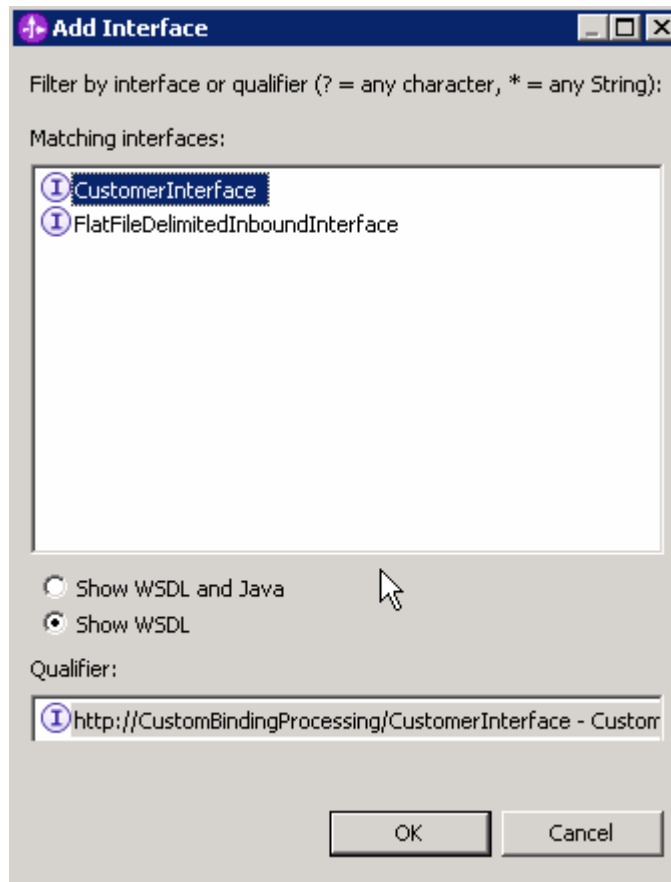
- 1) From Business Integration window, Expand **CustomBindingProcessing > Assembly Diagram** and double-click on **CustomBindingProcessing** to open the in Assembly diagram editor

IBM WebSphere Adapter 6.0.2 – Lab Exercise

- \_\_\_ b. From the diagram window, click on the icon for **Component (with no implementation type)**. select **Java** component from the context menu icon then click in the Assembly Diagram's empty space to drop it there
- \_\_\_ c. Rename Component1 to **ProcessComponent**
- \_\_\_ d. Right-click on **ProcessComponent** and select **Add > Interface** from the context menu



- \_\_\_ e. Select the radio button **show WSDL**
- \_\_\_ f. Select **CustomerInterface** from the Matching interfaces section



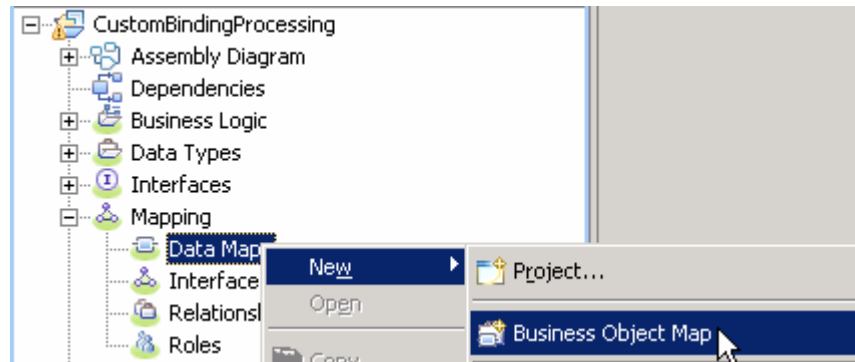
- \_\_\_ g. Click **OK**
- \_\_\_ h. Save (**Ctrl+S**) your changes and close the Assembly Diagram: CustomBindingProcessing

## 2.3. Create data and interface maps

\_\_\_ 1. Create FFDelimitedAddressToAddress map

\_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**

\_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**



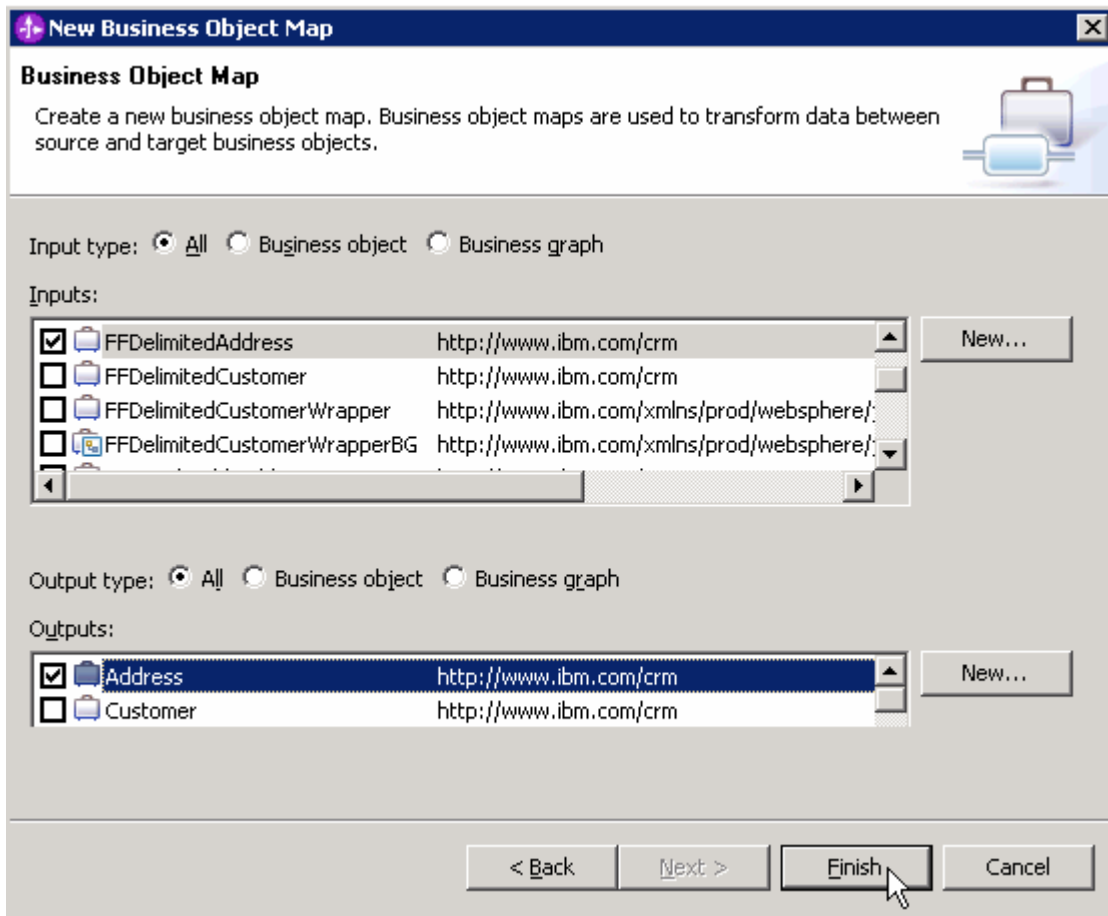
\_\_\_ c. From the Business Object Map screen, for the **Name** field, enter **FFDelimitedAddressToAddress**

\_\_\_ d. Click **Next**

\_\_\_ e. Select the following from the Business Object Map screen

1) Under **Inputs** section, select the check box for **FFDelimitedAddress**

2) Under **Outputs** section select the check box for **Address**

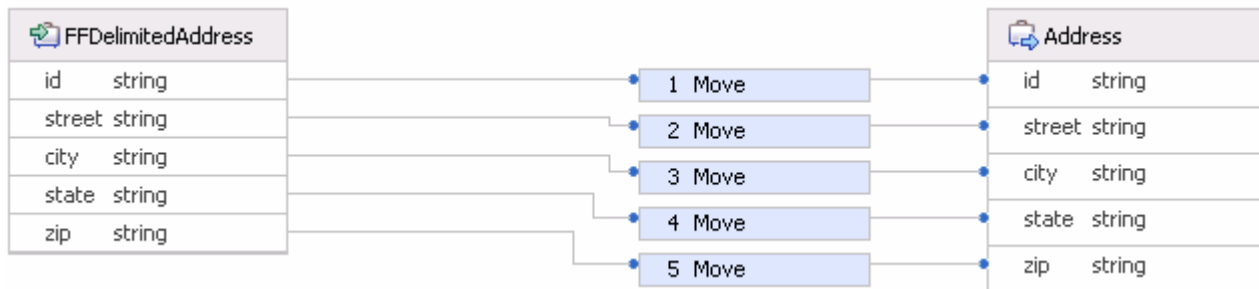


3) Click **Finish**

The FFDelimitedAddressToAddress map will be opened for transformation. Ignore the error that you will see at this point. It will be resolved in the following step.

\_\_\_ f. Add a connection from the **id field** of FFDelimitedAddress business object to the **id field** of Address business object

To perform this step, hover your mouse over the **id field** of FFDelimitedAddress business object and click on **Add connection** then click on the **id field** of Address business object. Using the same method, add the following connections:



\_\_\_ g. Save (**Ctrl +S**) your changes and close the FFDelimitedAddressToAddress map



---

**Note:** You should not see any errors at this point. Ignore any exceptions.

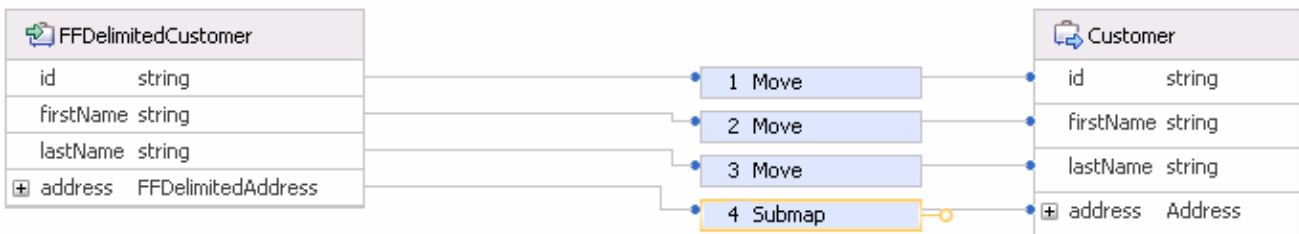
---

\_\_\_ 2. Create FFDelimitedCustomerToCustomer map

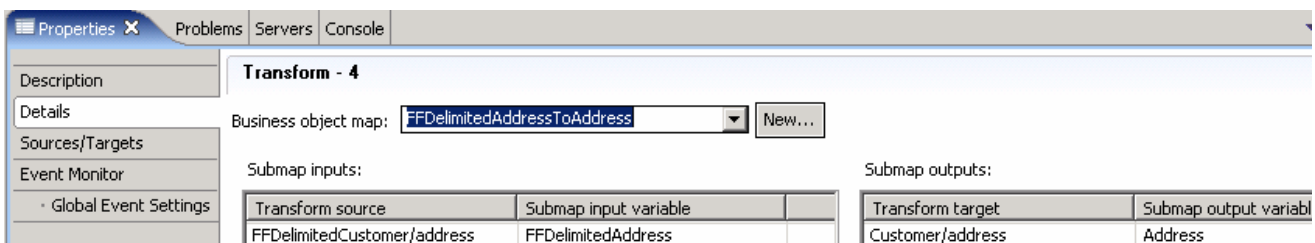
- \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
- \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
- \_\_\_ c. For the Name field, enter **FFDelimitedCustomerToCustomer** and click **Next**
- \_\_\_ d. From the Business Object Map screen, select the following
  - 1) Under **Inputs**, select the check box for **FFDelimitedCustomer**
  - 2) Under **Outputs**, select the check box for **Customer**
  - 3) Click **Finish**

The FFDelimitedCustomerToCustomer map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

- \_\_\_ e. Add the following connections



- \_\_\_ f. Click on **4 Submap**
- \_\_\_ g. From the bottom window, click the **Properties** tab.
- \_\_\_ h. Click on **Details**
- \_\_\_ i. For Business object map field, select **FFDelimitedAddressToAddress** from the drop down menu



- \_\_\_ j. Save (**Ctrl+S**) your changes and close the FFDelimitedCustomerToCustomer map

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

---

\_\_\_ 3. Create FFDelimitedCustomerWrapperBGToCustomer map

- \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**

- \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
- \_\_\_ c. For the Name field, enter **FFDelimitedCustomerWrapperBGToCustomer** and click **Next**
- \_\_\_ d. From the Business Object Map screen, select the following
  - 1) Under **Inputs**, select the check box for **FFDelimitedCustomerWrapperBG**
  - 2) Under **Outputs**, select the check box for **Customer**
  - 3) Click **Finish**

The FFDelimitedCustomerWrapperBGToCustomer map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

- \_\_\_ e. Add the following connection



- \_\_\_ f. Click on **1 Submap**
- \_\_\_ g. From the bottom window, click the **Properties** tab.
- \_\_\_ h. Click on **Details**
- \_\_\_ i. For Business object map field, select **FFDelimitedCustomerToCustomer** from the drop down menu
- \_\_\_ j. Save (**Ctrl+S**) your changes and close the FFDelimitedCustomerWrapperBGToCustomer map

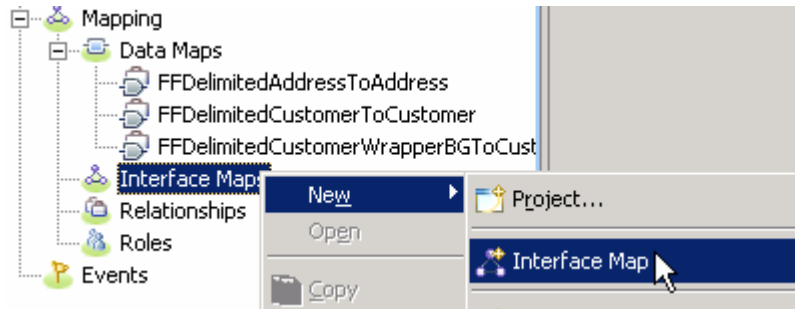
---

**Note:** You should not see any errors at this point. Ignore any exceptions.

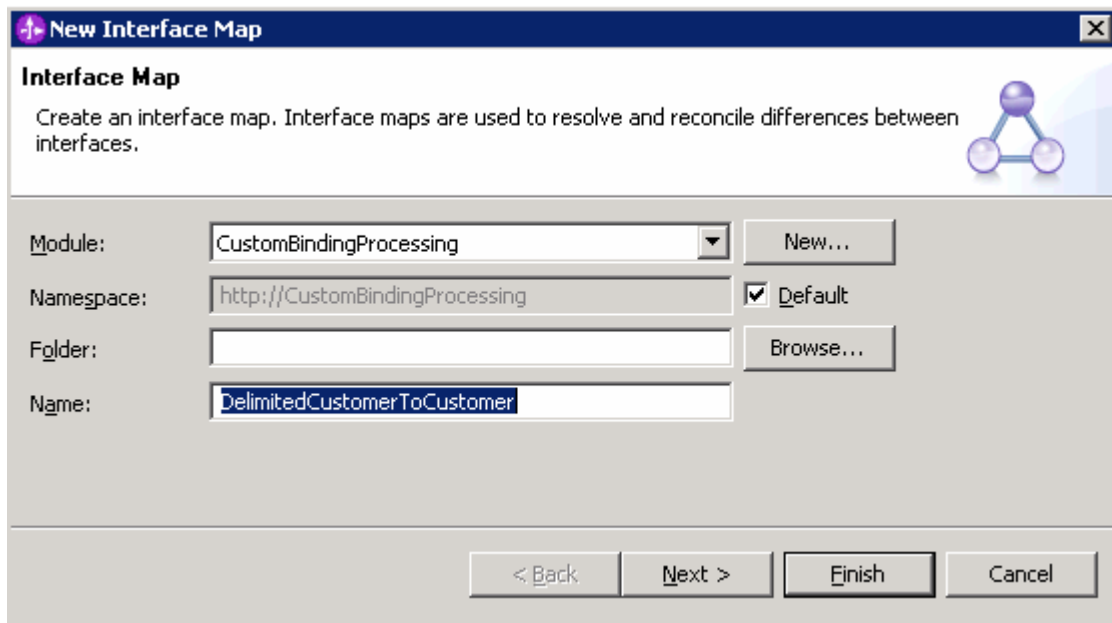
---

- \_\_\_ 4. Create DelimitedCustomerToCustomer Interface Map
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Interface Maps** and select **New > Interface Map**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

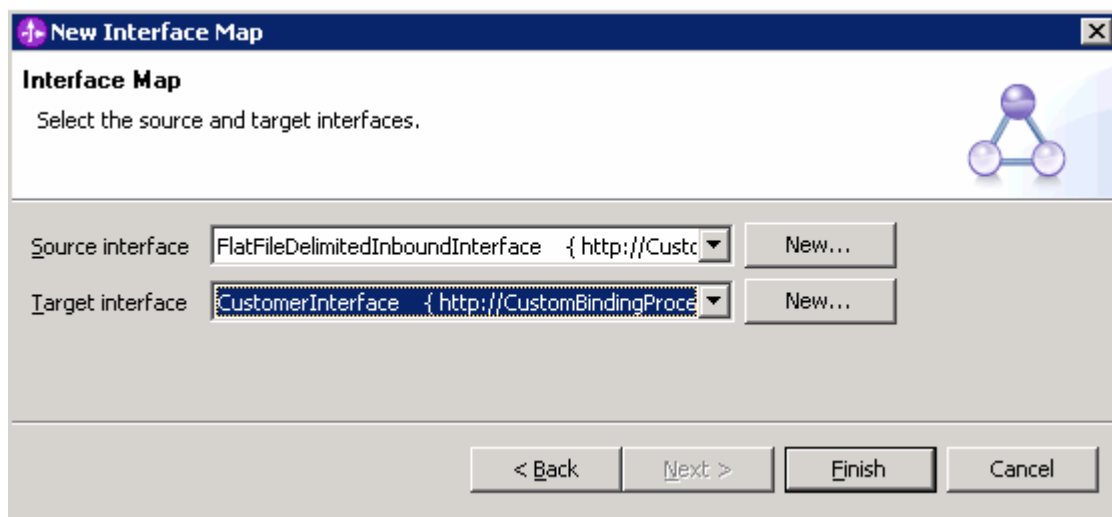


\_\_ c. For the **Name** field, enter **DelimitedCustomerToCustomer**



\_\_ d. Click **Next**

\_\_ e. Select **FlatFileDelimitedInboundInterface** for **Source interface** and **CustomerInterface** for **Target interface** and click **Finish**



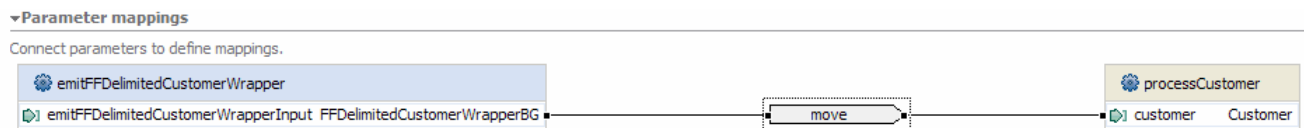
DelimitedCustomerToCustomer Interface map will be opened for mapping. Ignore any errors at this point. They will be resolved in the following step.

- \_\_\_ f. Create mapping from **emitFFDelimitedCustomerWrapper** field of FlatFileDelimitedInboundInterface to **processCustomer** field of CustomerInterface



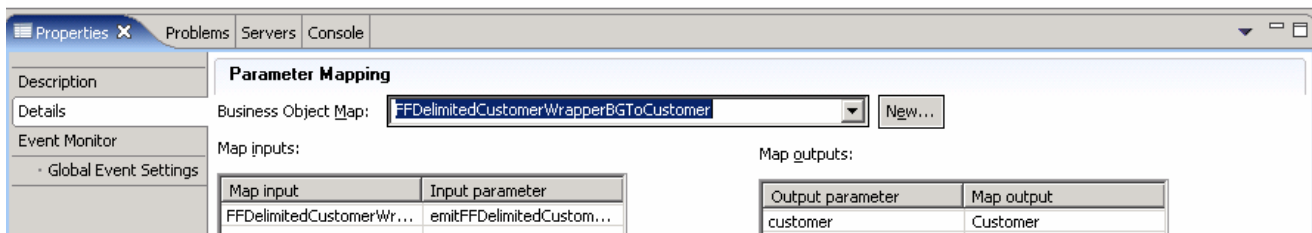
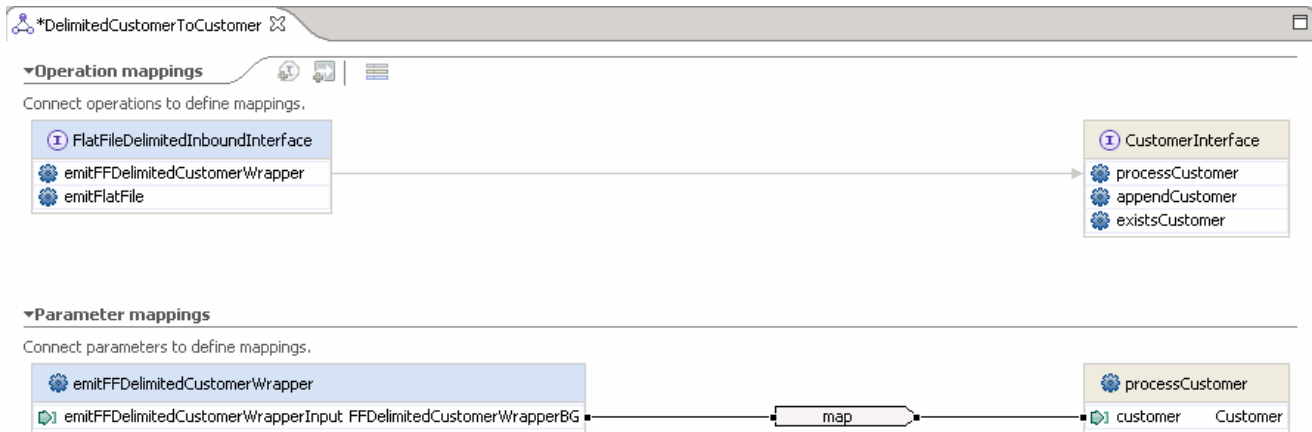
- \_\_\_ g. Click on the mapping line under Operation mappings to see the Parameter mappings

- \_\_\_ h. Create mapping from **emitFFDelimitedCustomerWrapperInput** to **customer**



- \_\_\_ i. Click on **move**
- \_\_\_ j. From **Properties** view, select **Description** then select **map** from the drop down menu for the **Parameter Mapping Type** field
- \_\_\_ k. From **Properties** view, select **Details** tab and then **FFDelimitedCustomerWrapperBGToCustomer** from the drop down menu for the **Business Object Map** field

IBM WebSphere Adapter 6.0.2 – Lab Exercise



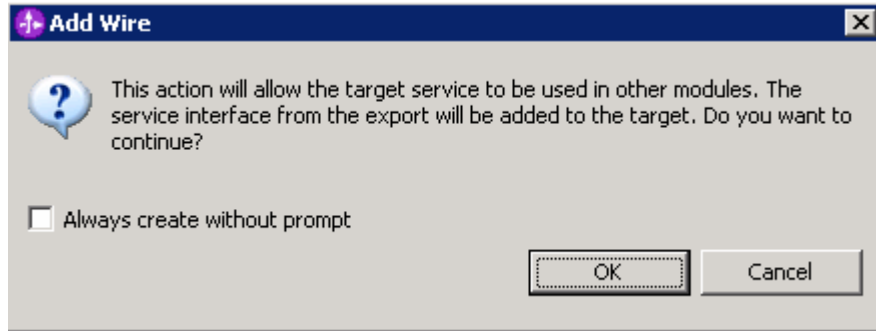
\_\_ I. Save (**Ctrl+S**) your changes and close the `DelimitedCustomerToCustomer` interface map window

---

**Note:** You should not see any errors at this point. Ignore any exceptions

---

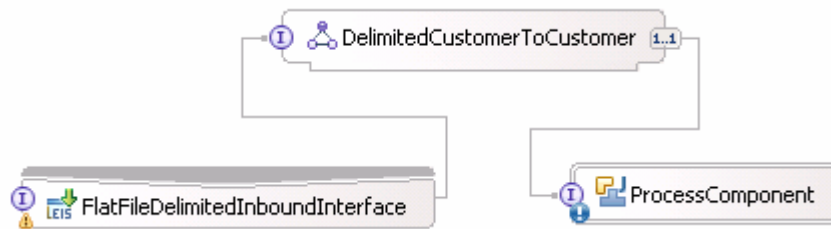
- \_\_\_ 5. Wire the components
  - \_\_\_ a. Open the Assembly diagram for **CustomBindingProcessing**
  - \_\_\_ b. Select **CustomBindingProcessing > Mapping > Interface Maps > DelimitedCustomerToCustomer** and drop it in the Assembly diagram
  - \_\_\_ c. Add a wire from `FlatFileDelimitedInboundInterface` to `DelimitedCustomerToCustomer`
    - 1) Click on **FlatFileDelimitedInboundInterface**, select **Add wire** and then click on **DelimitedCustomerToCustomer**
    - 2) Click **OK** for the following pop up window:



\_\_ d. Add a wire from DelimitedCustomerToCustomer to ProcessComponent

- 1) Click on **DelimitedCustomerToCustomer**, select **Add wire** and then click on **ProcessComponent**

You will see the following in the Assembly diagram



\_\_ e. Save (**Ctrl+S**) your changes and close the Assembly Diagram: CustomBindingProcessing

## 2.4. Run intermediate test

**Note:** Refer to the **Appendix: Troubleshooting Tips** at the end of this lab for any exceptions/errors that you might see while running the intermediate test.

- \_\_\_ 1. Generate Implementation for ProcessComponent
  - \_\_\_ a. From the **CustomBindingProcessing** assembly diagram, right-click on **ProcessComponent** and select **Generate Implementation**
  - \_\_\_ b. Select **default package** from the Generate Implementation window and click **OK**

The ProcessComponentImpl.java window will be opened.

- \_\_\_ c. Add the following to the processCustomer method:

```
public void appendCustomer(DataObject customer) {
    System.out.println("*****Inside processCustomer*****");
}
```

- \_\_\_ d. Save (**Ctrl+S**) your changes and close ProcessComponentImpl.java
  - \_\_\_ e. Save (**Ctrl+S**) your changes in the Assembly Diagram
- \_\_\_ 2. Start the WebSphere Process Server (if it is not already started)
    - \_\_\_ a. Select the **Servers** view
    - \_\_\_ b. Right-click on the row that contains **WebSphere Process Server V6.0** and select **Start** from the context menu
    - \_\_\_ c. Wait until the server status shows as **Started**
  - \_\_\_ 3. Add the project to the WebSphere Test Environment server
    - \_\_\_ a. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
    - \_\_\_ b. From the Add and Remove Projects window, select **CustomBindingProcessingApp** under Available projects panel and click **Add >**
    - \_\_\_ c. You will now see the **CustomBindingProcessingApp** added to the **Configured projects**
    - \_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server will be started in Debug mode if it is not already started before
  - \_\_\_ 4. Test the adapter application
    - \_\_\_ a. Copy a test file to the **<EVENT\_DIR>**. As the file is copied to the event directory, the same will be picked up and processed by the adapter and will be moved to the **<ARCHIVE\_DIR>**

**Note:** For your convenience, a test file named **Delimited.txt** file can be found in **<LAB\_FILES>DTF**.

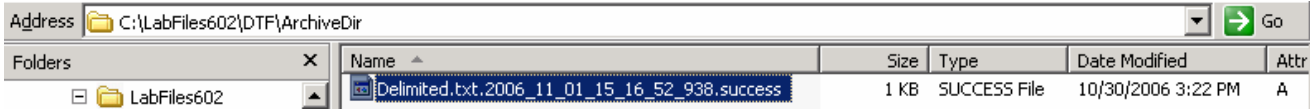
- \_\_\_ 5. Verify your test results

IBM WebSphere Adapter 6.0.2 – Lab Exercise

- \_\_\_ a. Check your test environment console (or SystemOut.log file) for the following successful message:

```
0000005c ResourceAdapt I com.ibm.j2ca.extension.eventmanagement.internal.EventSender sendEvent() CWYBS0505I: The event has been delivered.
0000005c SystemOut O *****Inside processCustomer*****
```

- \_\_\_ b. Now, 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



\_\_\_ 6. Restore the Sever Configuration

- \_\_\_ a. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu.
- \_\_\_ b. Select **CustomBindingProcessingApp** 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 configuration - NameValue

In this part, you will perform the following main tasks:

- You will use Enterprise Service Discovery (ESD) wizard to generate the FlatFileNameValueOutboundInterface, and also the artifacts –, ExistsResponse, ExistsResponseBG, FFNameValueCustomerWrapper, FFNameValueCustomerWrapperBG and so forth.
- You will create the following maps
  - Data maps – AddressToFFNameValueAddress, CustomerToFFNameValueCustomer, and CustomerToFFNameValueCustomerWrapperBG
  - Interface Map –CustomerToNameValueCustomer
- You will test the adapter for the NameValue Outbound scenario

---

### 3.1. Run ESD to create outbound interface and other artifacts

- \_\_\_ 1. Launch the Enterprise Service Discovery wizard
  - \_\_\_ a. From the top Menu, select **File > New > Enterprise Service Discovery**
- \_\_\_ 2. Select **IBM WebSphere Adapter for Flat Files (version 6.0.2)** from the 'CWYFF\_FlatFile' **Connector Project** if not selected by default and click **Next**
- \_\_\_ 3. Enter the following for the Configure Settings for Discovery Agent screen
  - \_\_\_ a. Click on Browser... next to Folder Name and select **<WORKSPACE>\CustomBindingProcessing**
  - \_\_\_ b. Select **UTF-8** from the drop down menu for the **Character Set**
  - \_\_\_ c. Select **text/namevalue** from the drop down menu for the **Content Type** and the **NameValueDataBinding** will automatically be selected as the **DataBinding Type**
  - \_\_\_ d. Enter the following for the DataBindingProperties
    - 1) TargetNameSpace: <http://www.ibm.com/crm>
  - \_\_\_ e. Click on **Show Advanced** to see the logging properties. Note the **Log file output location**

**Enterprise Service Discovery**

### Configure Settings for Discovery Agent

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

**Connection Configuration**

Folder Name:  Browse...

Character Set:

Content Type:

DataBinding type:

Specify BO Properties

**BIDI Properties**

Bidi transformation

Bidi ordering schema:

Text direction:

Bidi Symmetric Swapping

Bidi shaping:

Bidi numeric shaping:

**DataBinding properties**

TargetNamespace:

DataBindingClass:

**Logging options**

Log file output location:\*  Browse...

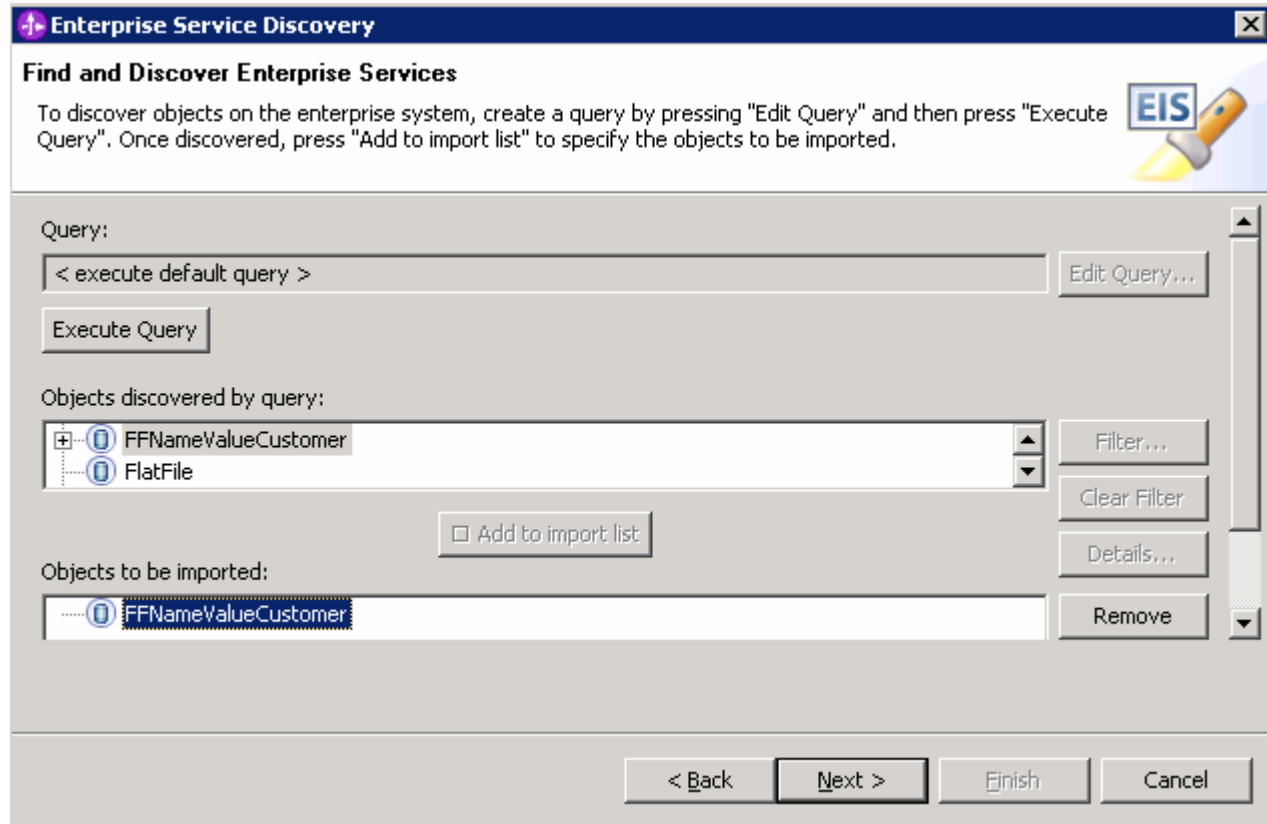
Logging Level:

\_\_\_ f. Click **Next**

\_\_\_ 4. Find and discover the enterprise services. In this step you will select the business objects and services to be used with the adapter

\_\_\_ a. From the Enterprise Service Discovery window, click on **Execute Query** button. You will see all the business objects the you imported under **Objects discovered by query**

\_\_\_ b. Select **FFNameValueCustomer** business object and click **>>Add to import list** button. The Customer business object will now be displayed under the **Objects to be imported**

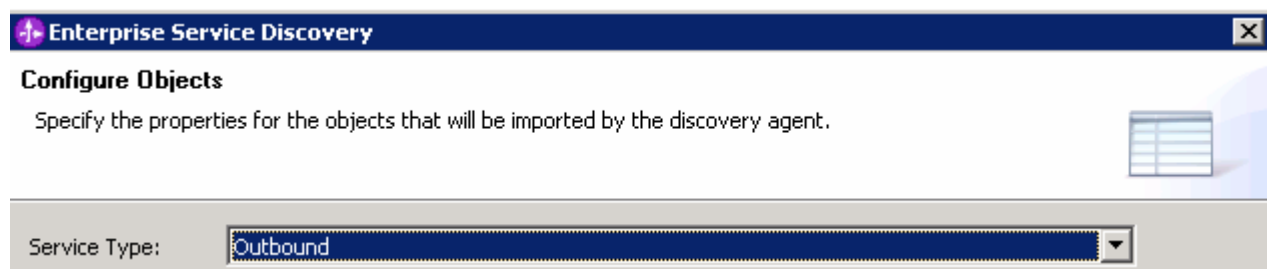


\_\_ c. Click **Next**

\_\_\_ 5. Configure the objects that will be imported by the discovery agent

\_\_ a. From the Configure objects window, select **Outbound** if not already selected from the dropdown menu for the **Service Type**.

\_\_ b. Click **Next**



\_\_\_ 6. Enter the following from the Generate Artifacts window

\_\_ a. Enter the following for **Properties for Interface**

1) For **Module** field, ensure that **CustomBindingProcessing** is selected

2) For **Name** field, enter **FlatFileNameValueOutboundInterface**

Properties for service

Module: CustomBindingProcessing New...

Namespace: http://CustomBindingProcessing/FlatFileNameValueOutbound

Use default namespace

Folder: Browse...

Name: \* FlatFileNameValueOutboundInterface

Description:

\_\_ b. Ensure that **Use discovered connection properties** radio button is selected

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

Use connection properties specified on server

Use discovered connection properties

\_\_ c. Enter the following for **FlatFile MCF Properties**

- 1) Output Directory: <DATA\_DIR>
- 2) Staging Directory: <STAGING\_DIR>

\_\_ d. Enter the following for **ResourceAdapterProperties**

- 1) LogFilename: C:\FFRA\DTF\NameValueOutlog.txt
- 2) TraceFilename: C:\FFRA\DTF\NameValueOuttrace.txt

Connection properties

FlatFile MCF Properties

Output Directory: C:\LabFiles602\DTF\DataDir

Output File Name: NameValue.txt

Staging Directory:

ResourceAdapter Properties

Logging and Tracing

Adapter ID [String]: \* ResourceAdapter

Log file size [Integer]: 0

Log file name [String]: C:\FFRA\DTF\NameValueOutlog.txt

Log Files [Integer]: 1

Trace file size [Integer]: 0

Trace file name [String]: C:\FFRA\DTF\NameValueOuttrace.txt

Trace files [Integer]: 1

\_\_ e. Click **Finish**

- \_\_\_ 7. From Business Integration window, expand **CustomBindingProcessing > Data Types**
  - \_\_\_ a. Right-click on **FFNameValueCustomer** and select **Open With > Text Editor** to open it
  - \_\_\_ b. Scroll down to the place where it shows the error and delete the following

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xi:include schemaLocation="FFNameValueAddress.xsd" />
<include schemaLocation="FFNameValueAddress.xsd" />
```

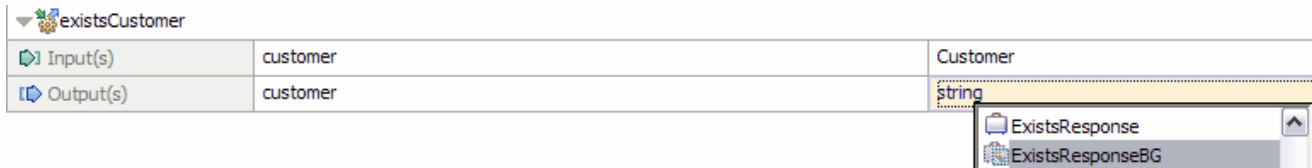
- \_\_\_ c. Save (**Ctrl+S**) your changes and close FFNameValueCustomer business object

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

---

- \_\_\_ 8. Edit CustomerInterface with the ExistsResponseBG output
  - \_\_\_ a. Expand CustomBindingProcessing > Interfaces and double-click on CustomerInterface to open it
  - \_\_\_ b. Under the existsCustomer operation, for Output Type, click on string and select ExistsResponseBG from the context menu



- \_\_\_ c. Save (**Ctrl+S**) your changes and close CustomerInterface

- \_\_\_ 9. Regenerate Implementation for ProcessComponent
  - \_\_\_ a. Expand **CustomBindingProcessing** and double click on **Assembly Diagram** to open it in Assembly editor
  - \_\_\_ b. Right-click on **ProcessComponent** and select **Regenerate Implementation**

A pop-up window will come up. Click **OK** to replace.

The implementation will automatically be regenerated and the ProcessComponentImpl.java will be opened.

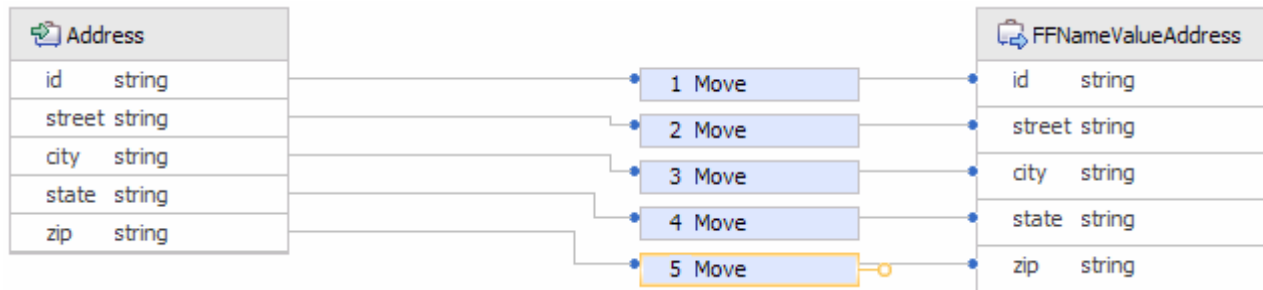
- \_\_\_ c. Close the **ProcessComponentImpl.java** and save (**Ctrl+S**) the changes to the Assembly diagram

## 3.2. Create data and interface maps

- \_\_\_ 1. Create AddressToFFNameValueAddress map
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
  - \_\_\_ c. From the Business Object Map screen, enter **AddressToFFNameValueAddress** for **Name** and click **Next**
  - \_\_\_ d. Select the following from the Business Object Map screen
    - 1) Under Inputs section, select the check box for **Address**
    - 2) Under Outputs section select the check box for **FFNameValueAddress**
    - 3) Click **Finish**

The AddressToFFNameValueAddress map will be opened for transformation. Ignore the error that you will see at this point. It will be resolved in the following step.

- \_\_\_ e. Add the following connections



- \_\_\_ f. Save (**Ctrl +S**) your changes and close the AddressToFFNameValueAddress map

---

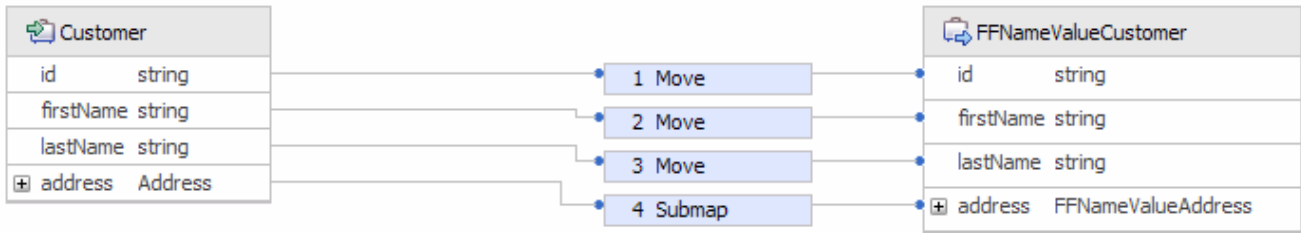
**Note:** You should not see any errors at this point. Ignore any exceptions.

---

- \_\_\_ 2. Create CustomerToFFNameValueCustomer map
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
  - \_\_\_ c. Enter **CustomerToFFNameValueCustomer** for the **Name** field and click **Next**
  - \_\_\_ d. From the Business Object Map screen, select the following
    - 1) Under Inputs section, select the check box for **Customer**
    - 2) Under Outputs section select the check box for **FFNameValueCustomer**
    - 3) Click **Finish**.

The CustomerToFFNameValueCustomer map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

\_\_\_ e. Add the following connections

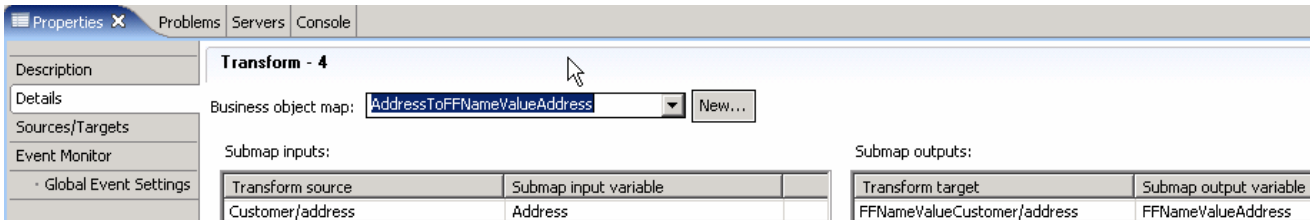


\_\_\_ f. Click on **4 Submap**

\_\_\_ g. From the bottom window, click the **Properties** tab.

\_\_\_ h. Click on **Details**

\_\_\_ i. For Business object map field, select **AddressToFFNameValueAddress** from the drop down menu



\_\_\_ j. Save (**Ctrl+S**) your changes and close the CustomerToFFNameValueCustomer map

**Note:** You should not see any errors at this point. Ignore any exceptions.

\_\_\_ 3. Create CustomerToFFNameValueCustomerWrapperBG map

\_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**

\_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**

\_\_\_ c. Enter **CustomerToFFNameValueCustomerWrapperBG** for the **Name** field and click **Next**

\_\_\_ d. From the Business Object Map screen, select the following

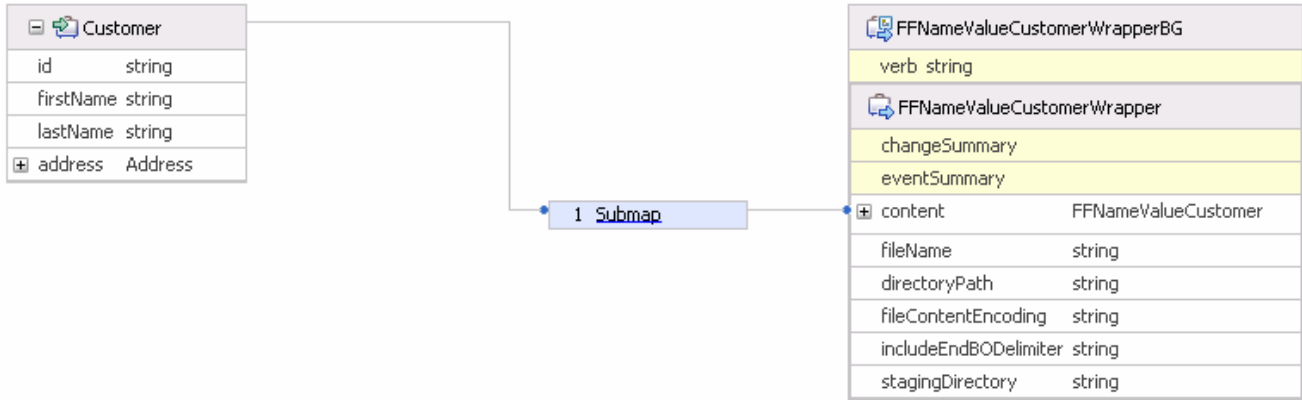
- 1) Under **Inputs** section select the check box for **Customer**
- 2) Under **Outputs** select the check box for **FFNameValueCustomerWrapperBG**
- 3) Click **Finish**

The CustomerToFFNameValueCustomerWrapperBG map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

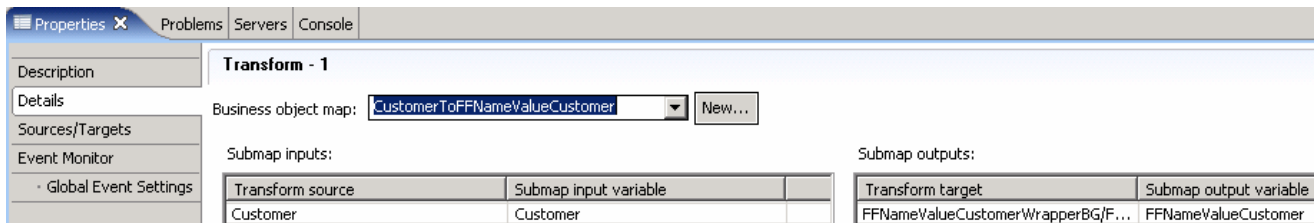
\_\_\_ e. Add the following connection



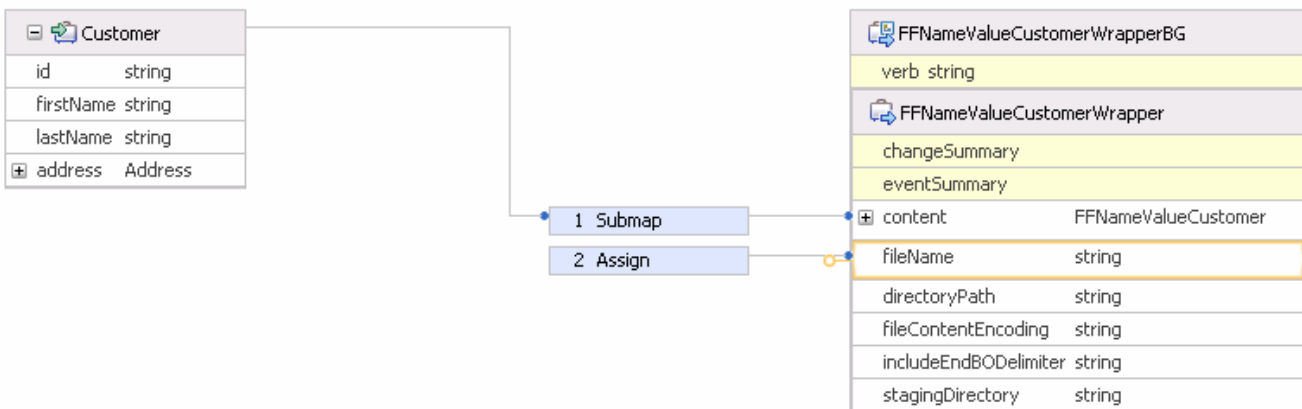
IBM WebSphere Adapter 6.0.2 – Lab Exercise



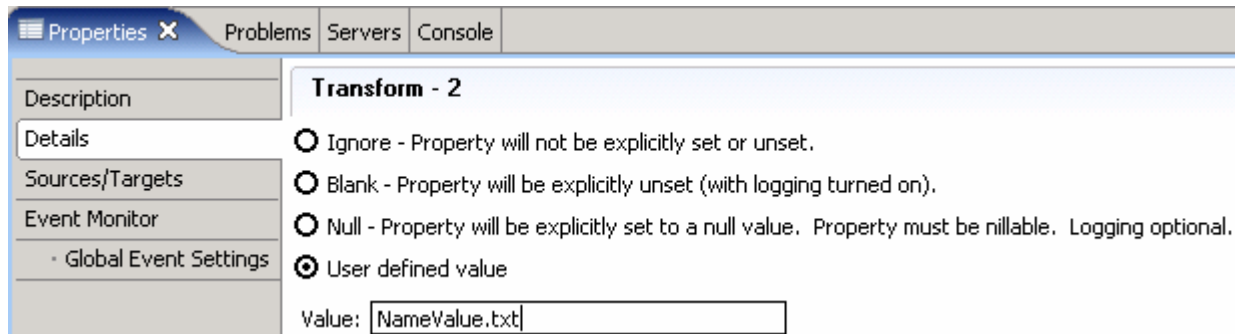
- \_\_\_ f. Click on **1 Submap**
- \_\_\_ g. From the bottom window, click the **Properties** tab.
- \_\_\_ h. Click on **Details**
- \_\_\_ i. For Business object map field, select **CustomerToFFNameValueCustomer** from the drop down menu



- \_\_\_ j. On the transformations, right-click on **fileName** and select **Create Transform > Assign** from the context menu



- \_\_\_ k. Ensure that **Assign** is selected and then click on **Details** under the **Properties** view from the bottom window
- \_\_\_ l. Enter **NameValue.txt** for **Value**



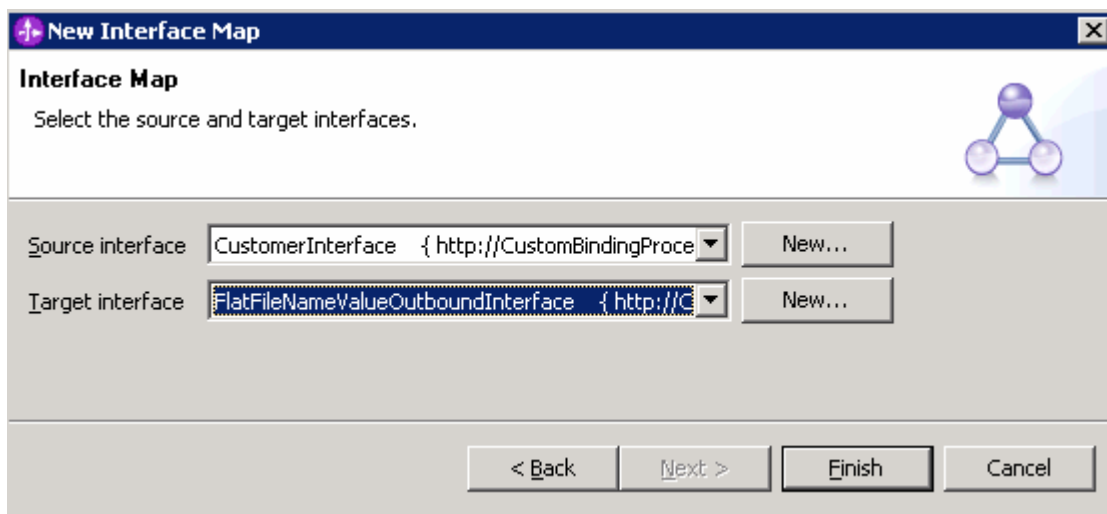
\_\_\_ m. Save (**Ctrl+S**) your changes and close the CustomerToFFNameValueCustomerWrapperBG map

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

---

- \_\_\_ 4. Create CustomerToNameValueCustomer Interface Map
- \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Interface Maps** and select **New > Interface Map**
  - \_\_\_ c. Enter **CustomerToNameValueCustomer** for the **Name** and click **Next**
  - \_\_\_ d. For **Source interface** field, select **CustomerInterface**
  - \_\_\_ e. For **Target interface** field, select **FlatFileNameValueOutboundInterface**
  - \_\_\_ f. Click **Finish**

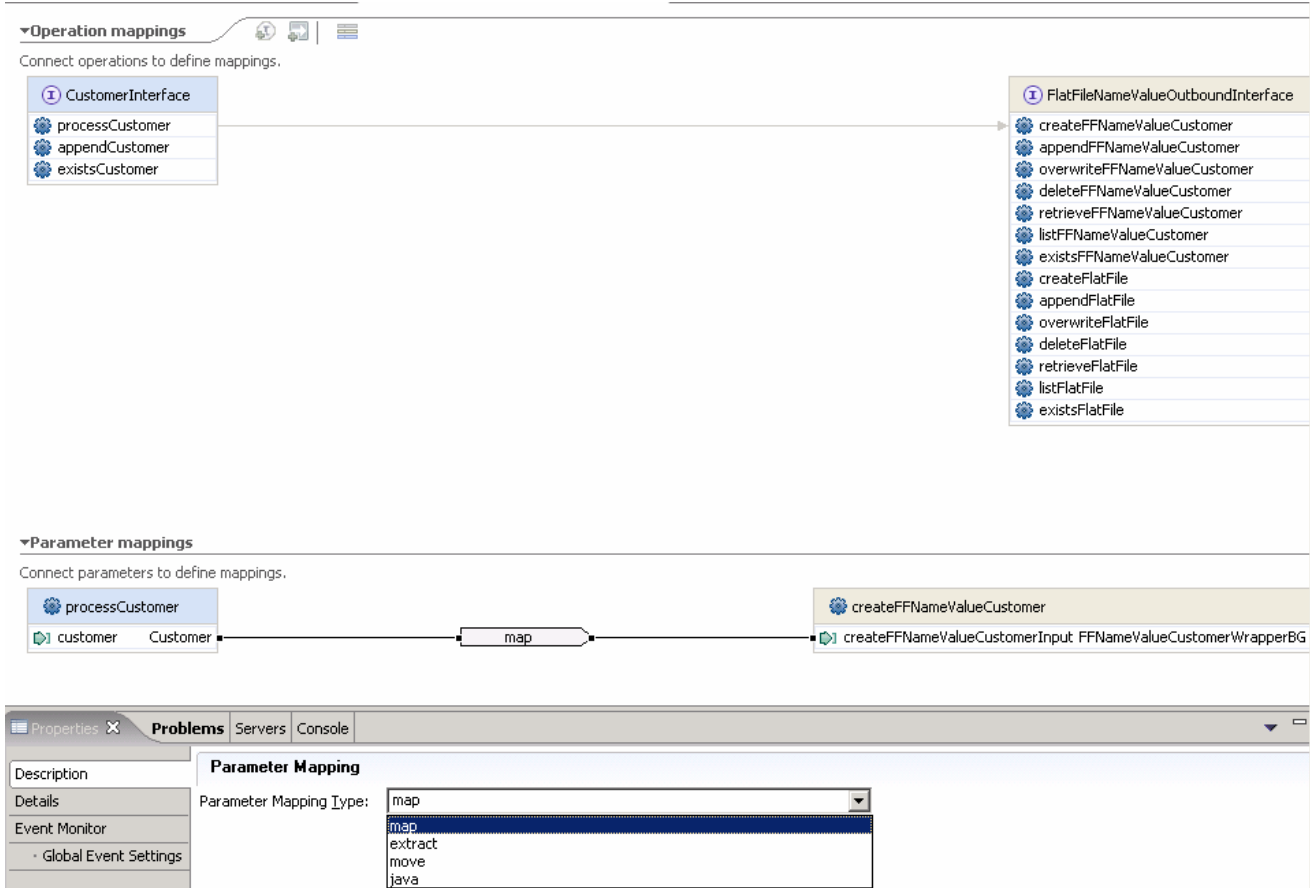


CustomerToNameValueCustomer Interface map will be opened for mapping. Ignore any errors at this point. They will be resolved in the following step.

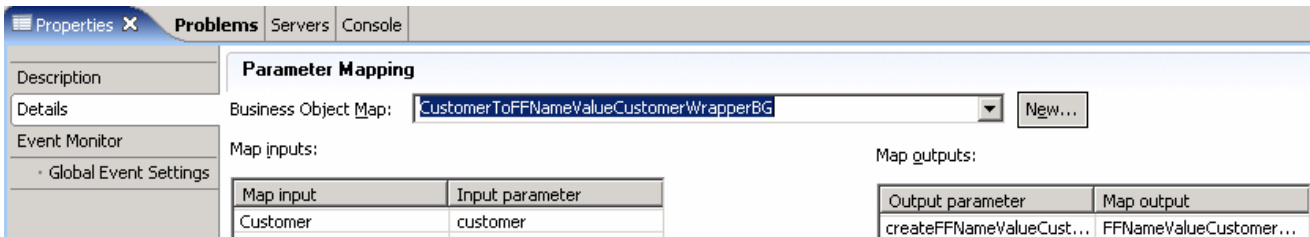
- \_\_\_ g. Create a map from processCustomer to createFFNameValueCustomer
  - 1) Click on **processCustomer**, select **Create operation mapping** and then click on **createFFNameValueCustomer**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

- 2) Click on the above mapping line and you will see **Parameter mappings** for it
- 3) Create a mapping from **Customer** to **createFFNameValueCustomerInput** under **Parameter mappings**
- 4) Click on **move** and from **Properties** view, select **Description** and then select **map** from the drop down menu for the Parameter Mapping Type field



- 5) From **Properties** view, select **Details** tab and then select **CustomerToFFNameValueCustomerWrapperBG** from the drop down menu for the **Business Object Map** field

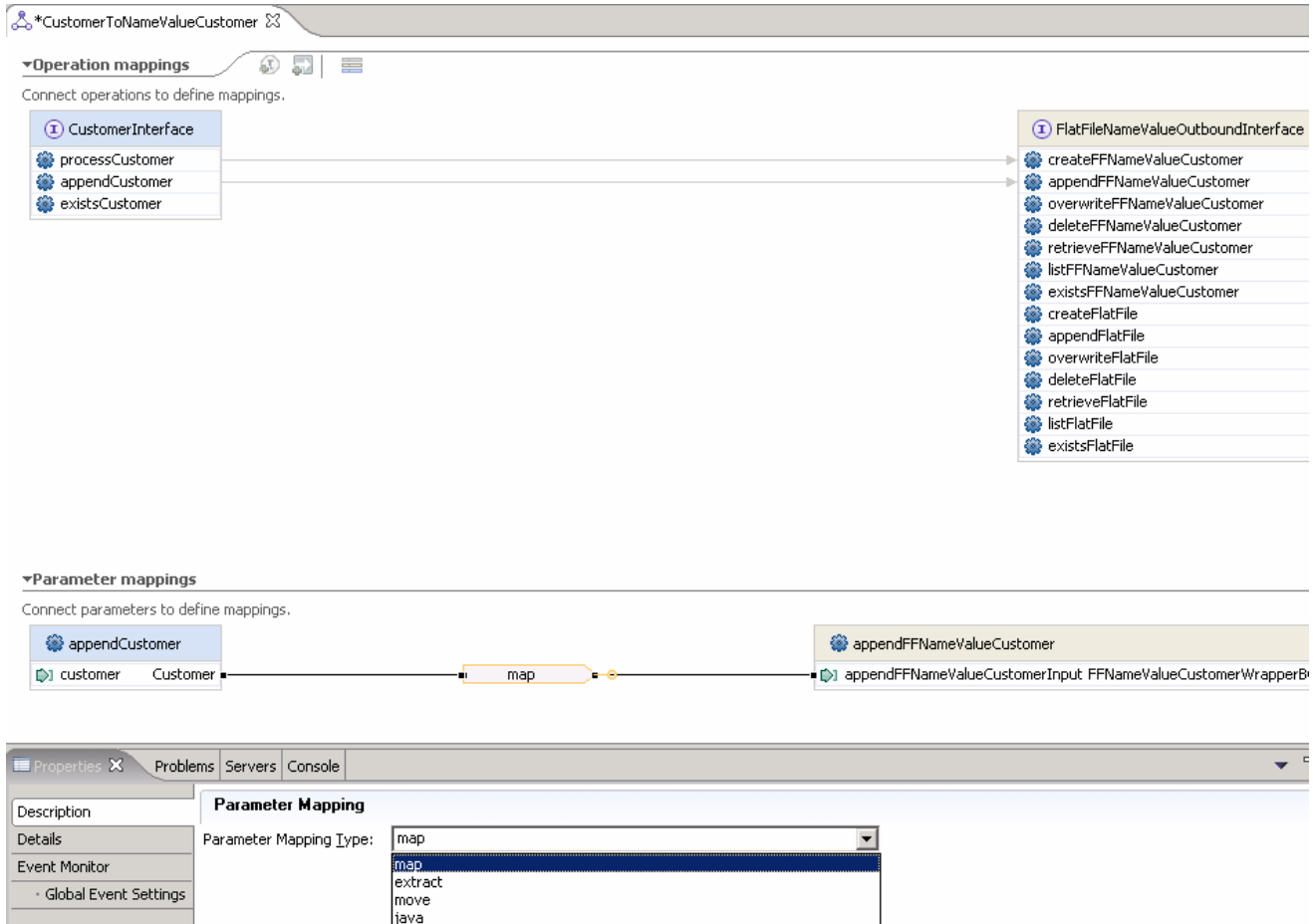


- \_\_\_ h. Create an operation mapping from **appendCustomer** to **appendFFNameValueCustomer**
- 1) Click on **appendCustomer**, select **Create operation mapping** and then click on **appendFFNameValueCustomer**
  - 2) Click on the above mapping line and you will see **Parameter mappings** for it

IBM WebSphere Adapter 6.0.2 – Lab Exercise

3) On Parameter mappings section, create a mapping from **Customer** to **appendFFNameValueCustomerInput**

4) Click on **move** and from **Properties** view, select **Description** and then select **map** from the drop down menu for the **Parameter Mapping Type** field

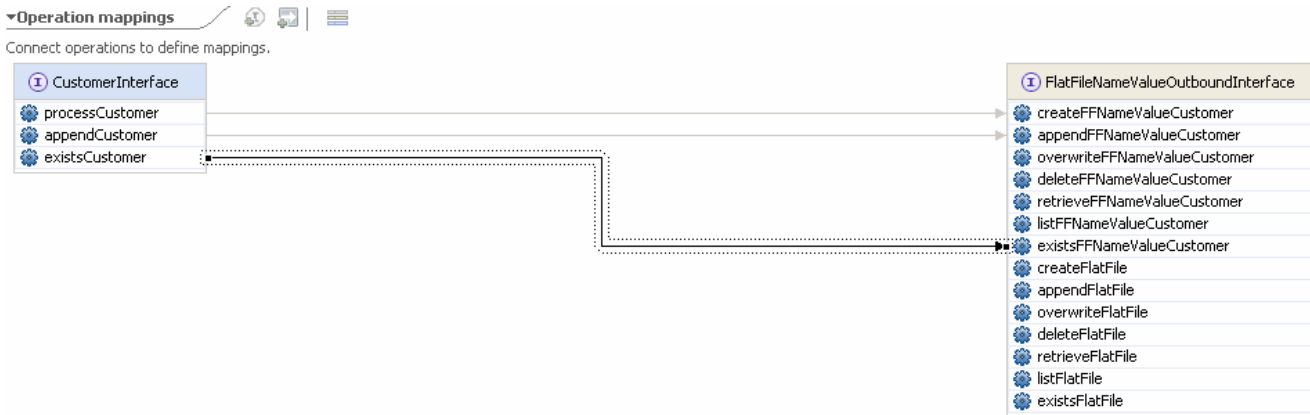


5) From **Properties** view, select **Details** tab and then select **CustomerToFFNameValueCustomerWrapperBG** from the drop down menu for the **Business Object Map** field

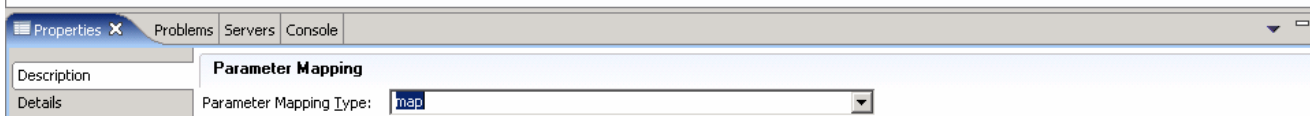
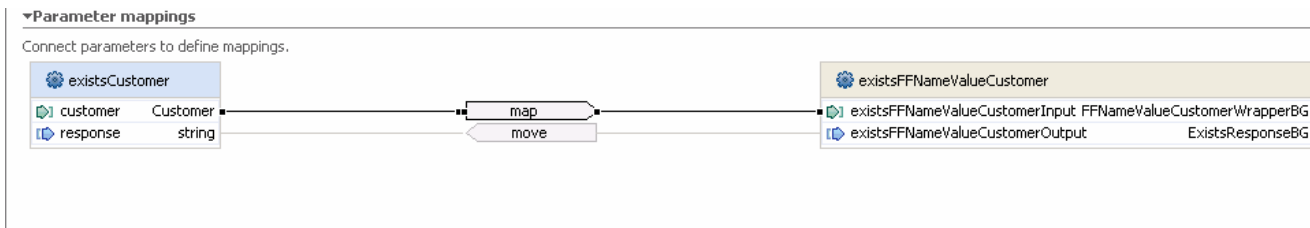
\_\_\_ i. Create a map from existsCustomer to existsFFNameValueCustomer

1) Click on **existsCustomer**, select **Create mapping** and then click on **existsFFNameValueCustomer**

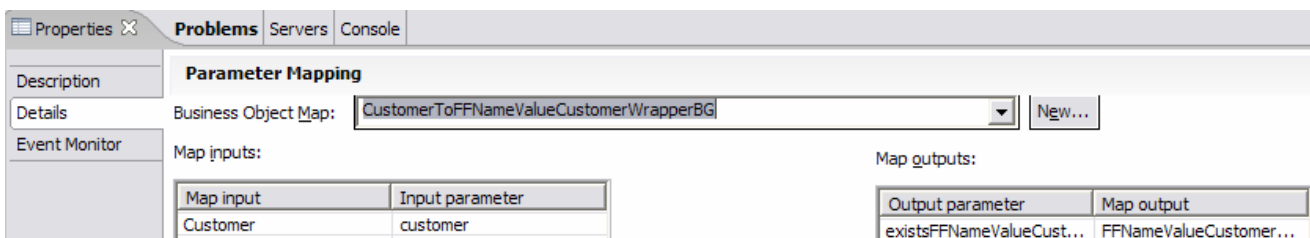
IBM WebSphere Adapter 6.0.2 – Lab Exercise



- 2) Click on the above mapping line and you will see **Parameter mappings** for it
- 3) Create mapping the following mappings under **Parameter mappings**
  - a) from **Customer** to **existsFFNameValueCustomerInput**
  - b) from **existsFFNameValueCustomerOutput** to **ExistsResponseBG**
- 4) Click on **move**, connecting from Customer to existsFFNameValueCustomerInput, and from **Properties** view, select **Description** and then select **map** from the drop down menu for the **Parameter Mapping Type** field



- 5) From **Properties** view, select **Details** tab and then select **CustomerToFFNameValueCustomerWrapperBG** from the drop down menu for the **Business Object Map** field



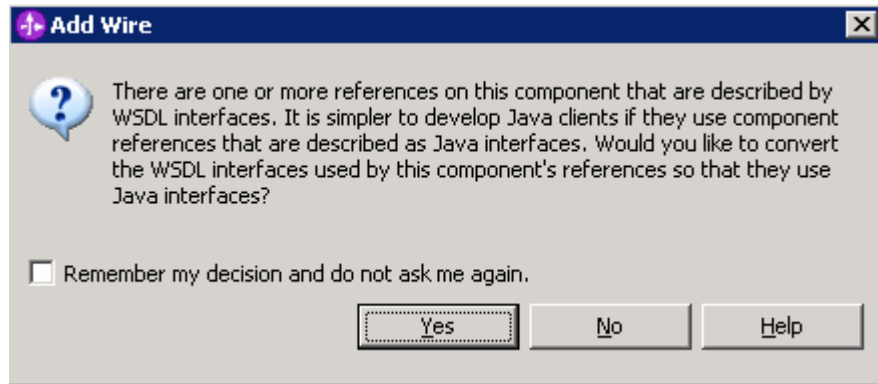
\_\_\_ j. Save (**Ctrl+S**) your changes and close the CustomerToNameValueCustomer interface map window

---

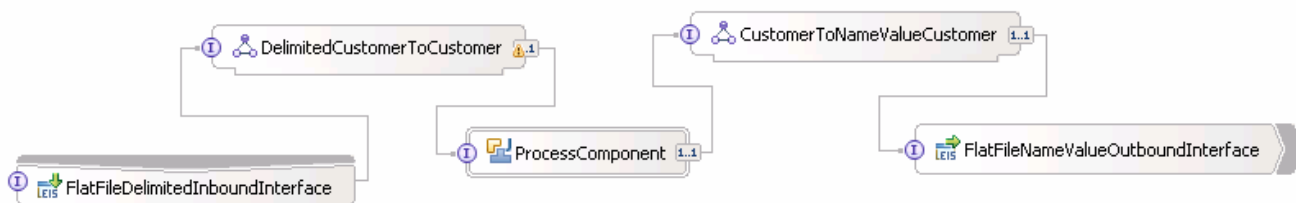
**Note:** You should not see any errors at this point. Ignore any exceptions.

---

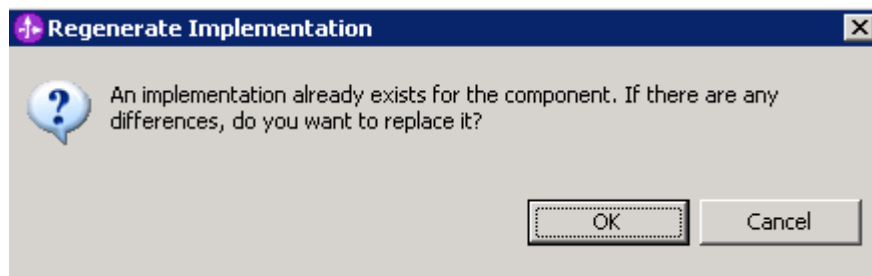
- \_\_\_ 5. Wire the components
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing** and double-click on **CustomBindingProcessing** to open it in Assembly Editor
  - \_\_\_ b. Select **CustomBindingProcessing > Mapping > Interface Maps > CustomerToNameValueCustomer** and drop it in the Assembly diagram
  - \_\_\_ c. Add a wire from **ProcessComponent** to **CustomerToNameValueCustomer**. Click **OK** for the following pop up window:
  - \_\_\_ d. Click **Yes** for the following window



- \_\_\_ e. Add a wire from **CustomerToNameValueCustomer** to **FlatFileNameValueOutboundInterface**
- You will see the following in the Assembly diagram



- \_\_\_ 6. Save (**Ctrl+S**) your changes on Assembly diagram
- \_\_\_ 7. Regenerate Implementation for ProcessComponent
  - \_\_\_ a. Right-click on **ProcessComponent** and select **Regenerate Implementation** from the context menu. This will automatically regenerate implementation. Click **OK** to the pop-up box



The **ProcessComponentImpl.java** will be opened for you for any required code changes. For this part, you do not have to add any code. You can see the added CustomerInterface method which is generated automatically.

\_\_\_ b. Close the **ProcessComponentImpl.java** window

\_\_\_\_ 8. Save (**Ctrl+S**) your changes and close the Assembly diagram

---

### 3.3. Run intermediate test

---

**Note:** Refer to the **Appendix: Troubleshooting Tips** at the end of this lab for any exceptions/errors that you might see while running the intermediate test.

---

- \_\_\_ 1. Start WebSphere Process Server test environment and add the **CustomBindingProcessingApp** as shown in **Part 2.4 Run Intermediate Test**

Wait for the project to be added to the server and the application to start. The server will be started in Debug mode if it is not already started before.

- \_\_\_ 2. Test the adapter application
- \_\_\_ a. From the Business Integration perspective, expand **CustomBindingProcessing** and double-click on CustomBindingProcessing to open it in assembly editor
  - \_\_\_ b. right-click on the **FlatFileNameValueOutboundInterface** and select **Test Component**

The FlatFileNameValueOutboundInterface\_Test will be opened

- \_\_\_ c. Under **Detailed Properties**, for the **Operation** field, select **createFFNameValueCustomer** from the drop down menu
- \_\_\_ d. Fill out the fields for Initial request parameters:
  - 1) For content and address fields enter some data of your choice such as shown on the image below
  - 2) fileName: **NameValue.txt**



IBM WebSphere Adapter 6.0.2 – Lab Exercise

Events

Invoke

General Properties

Detailed Properties

Configuration: Default Module Test

Module: CustomBindingProcessing

Component: FlatFileNameValueOutboundInterface

Interface: FlatFileNameValueOutboundInterface

Operation: createFFNameValueCustomer

Initial request parameters

Name	Type	Value
createFFNameValueCustomerInput	FFNameValueCustomerWrapperBG	
verb	string	<null>
FFNameValueCustomerWrapper	FFNameValueCustomerWrapper	
content	FFNameValueCustomer	
id	string	123
firstName	string	John
lastName	string	Doe
address	FFNameValueAddress	
id	string	11501
street	string	Burnet Rd
city	string	Austin
state	string	TX
zip	string	78758
fileName	string	NameValue.txt
directoryPath	string	
fileContentEncoding	string	
includeEndBODelimiter	string	
stagingDirectory	string	

\_\_ e. Click **Continue**

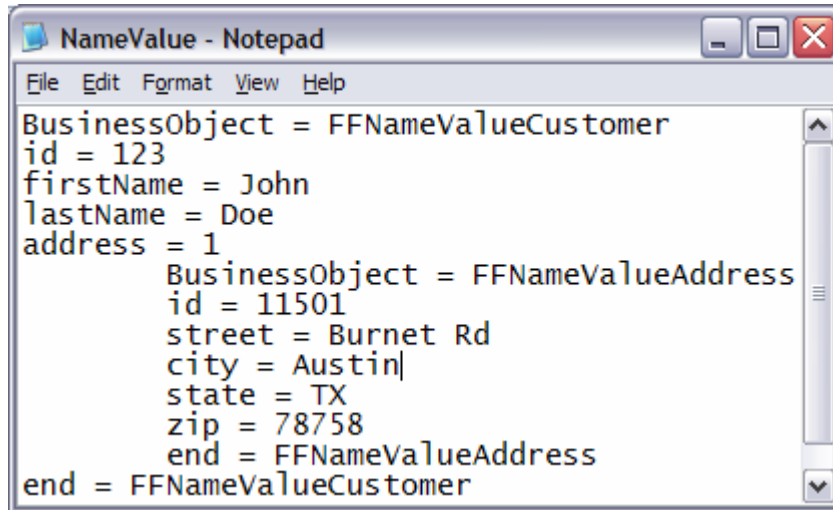
\_\_ f. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server V6.0** if not already selected, then click **Finish**

\_\_\_ 3. Verify the created file contents

\_\_ a. Check the subdirectory <DATA\_DIR> for the file **NameValue.txt**



\_\_ b. Double-click on **NameValue.txt** to open it and you should see the contents that you filled in the above steps:



```
BusinessObject = FFNameValueCustomer
id = 123
firstName = John
lastName = Doe
address = 1
    BusinessObject = FFNameValueAddress
    id = 11501
    street = Burnet Rd
    city = Austin
    state = TX
    zip = 78758
    end = FFNameValueAddress
end = FFNameValueCustomer
```

- \_\_\_ 4. Restore the Sever Configuration
  - \_\_\_ a. Close **FlatFileNameValueOutboundInterface\_Test** window and select **No** from the Save Resource pop up window
  - \_\_\_ b. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
  - \_\_\_ c. Select **CustomBindingProcessingApp** under Configured projects and click **< Remove**
  - \_\_\_ d. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished
  - \_\_\_ e. Delete the **NameValue.txt** file from **<DATA\_DIR>**

---

## Part 4: Outbound configuration - FixedWidth

In this part, you will perform the following main tasks:

- You will use Enterprise Service Discovery (ESD) wizard to generate the FlatFileFixedWidthOutboundInterface, and also the artifacts – FFFixedWidthCustomerWrapper, FFFixedWidthCustomerWrapperBG, and so forth.
- You will create the following maps
  - Data maps – AddressToFFFixedWidthAddress, CustomerToFFFixedWidthCustomer, and CustomerToFFFixedWidthCustomerWrapperBG
  - Interface Map –CustomerToFixedWidthCustomer
- You will test the adapter for the FixedWidth Outbound scenario

---

## 4.1. Run ESD to create outbound interface and other artifacts

- \_\_\_ 1. Launch the Enterprise Service Discovery wizard
  - \_\_\_ a. From the top Menu bar, select **File > New > Enterprise Service Discovery**
- \_\_\_ 2. Select **IBM WebSphere Adapter for Flat Files (version 6.0.2)** from the 'CWYFF\_FlatFile' **Connector Project** and click **Next**
- \_\_\_ 3. Enter the following for the **Configure Settings for Discovery Agent** screen
  - \_\_\_ a. Click on Browser... next to Folder Name and select **<WORKSPACE>\CustomBindingProcessing**
  - \_\_\_ b. Select **UTF-8** from the drop down menu for the **Character Set**
  - \_\_\_ c. For **content type** select **text/fixedwidth** from the drop down menu. **FixedWidth1DataBinding** will automatically be selected for **DataBinding Type**.
  - \_\_\_ d. Enter the following for the DataBindingProperties
    - 1) TargetNameSpace: **http://www.ibm.com/crm**
    - 2) PadCharacter: ~
    - 3) Field width: 15
  - \_\_\_ e. Scroll down and click on **Show Advanced** to see the logging properties. Note the **Log file output location**

**Enterprise Service Discovery**

**Configure Settings for Discovery Agent**

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

**Connection Configuration**

Folder Name: C:\LabFiles602\DTF\workspace\CustomBindingProcessing

Character Set: UTF-8

Content Type: text/fixedwidth

DataBinding type: FixedWidth1DataBinding

Specify BO Properties

**BiDi Properties**

Bidi transformation

Bidi ordering schema: Implicit

Text direction: LTR

BiDi Symmetric Swapping

Bidi shaping: Nominal

Bidi numeric shaping: Nominal

**DataBinding properties**

TargetNamespace: http://www.ibm.com/crm

FieldWidth: 15

PadCharacter: ~

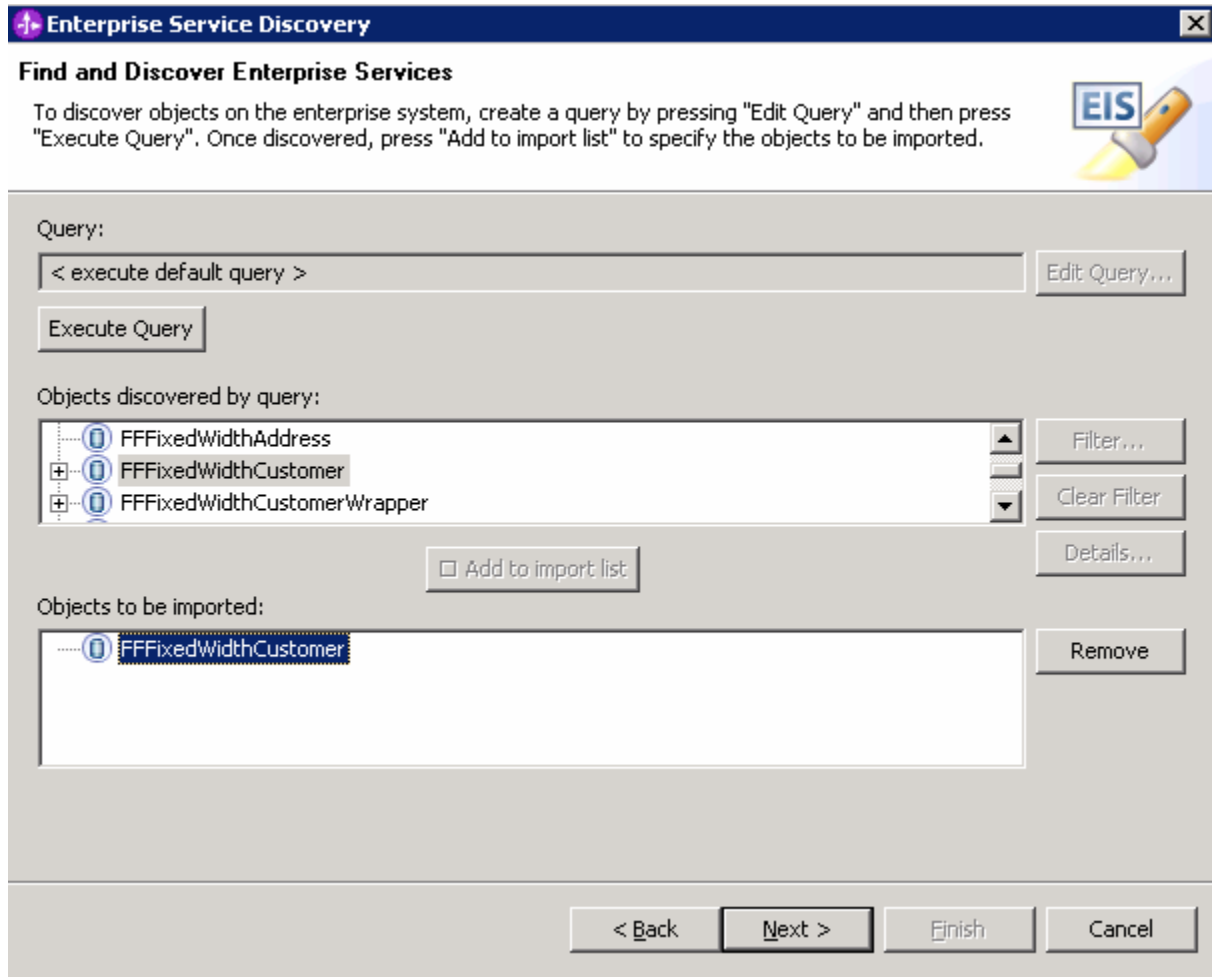
DataBindingClass: com.ibm.websphere.samples.databinding.FixedWidthEISDataBinding

\_\_\_ f. Click **Next**

\_\_\_ 4. Find and discover the enterprise services. In this step you will select the business objects and services to be used with the adapter

\_\_\_ a. From the Enterprise Service Discovery window, click on **Execute Query** button. You will see all the business objects that you imported under **Objects discovered by query**

\_\_\_ b. Select **FFFixedWidthCustomer** business object and click **>>Add to import list** button. The Customer business object will now be displayed under the **Objects to be imported**



\_\_\_ c. Click **Next**

\_\_\_ 5. Configure the objects that will be imported by the discovery agent

\_\_\_ a. From the Configure objects window, select **Outbound** (this is the default selection) from the dropdown menu for the **Service Type**

\_\_\_ b. Click **Next**

\_\_\_ 6. Enter the following from the Generate Artifacts window

\_\_\_ a. Enter the following for **Properties for Interface**

1) For **Module** field, ensure that **CustomBindingProcessing** is selected

2) For **Name** field enter **FlatFileFixedWidthOutboundInterface**

\_\_\_ b. Ensure that **Use discovered connection properties** radio button is selected

\_\_\_ c. Enter the following for **FlatFile MCF Properties**

1) Output Directory: **<DATA\_DIR>**

2) Staging Directory: **<STAGING\_DIR>**

## IBM WebSphere Adapter 6.0.2 – Lab Exercise

3) Output File Name: **FixedWidth.txt**

\_\_\_ d. Enter the following for **ResourceAdapterProperties**

1) LogFilename: **C:\FFRA\DTF\FWOutlog.txt**

2) TraceFilename: **C:\FFRA\DTF\FWOuttrace.txt**

Properties for service

Module: CustomBindingProcessing

Namespace: http://CustomBindingProcessing/FlatFileFixedWidthOutbound

Use default namespace

Folder:

Name: \* FlatFileFixedWidthOutboundInterface

Description:

---

Service operations

If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

---

Deployment properties

Deploy connector with module

J2C Authentication Data Entry:

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

Use connection properties specified on server

Use discovered connection properties

---

Connection properties

FlatFile MCF Properties

Output Directory: C:\LabFiles602\DTF\DataDir

Output File Name: FixedWidth.txt

Staging Directory: C:\LabFiles602\DTF\StagingDir

---

ResourceAdapter Properties

Logging and Tracing

Adapter ID [String]: \* ResourceAdapter

Log file size [Integer]: 0

Log file name [String]: C:\FFRA\DTF\FWOutlog.txt

Log Files [Integer]: 1

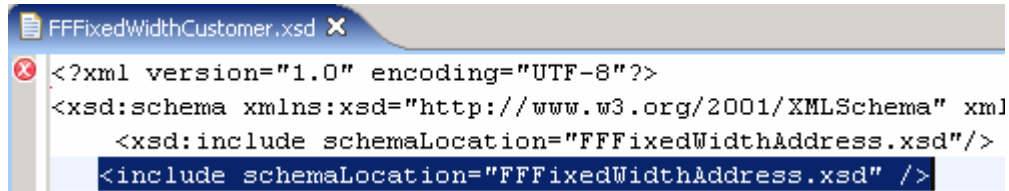
Trace file size [Integer]: 0

Trace file name [String]: C:\FFRA\DTF\FWOuttrace.txt

Trace files [Integer]: 1

\_\_\_ e. Click **Finish**

- \_\_\_ 7. From Business Integration window, expand **CustomBindingProcessing > Data Types**
- \_\_\_ a. Right-click on **FFFixedWidthAddress** and select **Open With > Text Editor** to open it
- \_\_\_ b. Scroll down to the place where it shows the error and delete the following:



```
FFFixedWidthCustomer.xsd X
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xml
  <xsd:include schemaLocation="FFFixedWidthAddress.xsd"/>
  <include schemaLocation="FFFixedWidthAddress.xsd" />
```

- \_\_\_ c. Save (**Ctrl+S**) your changes and close FFFixedWidthAddress BO

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

---

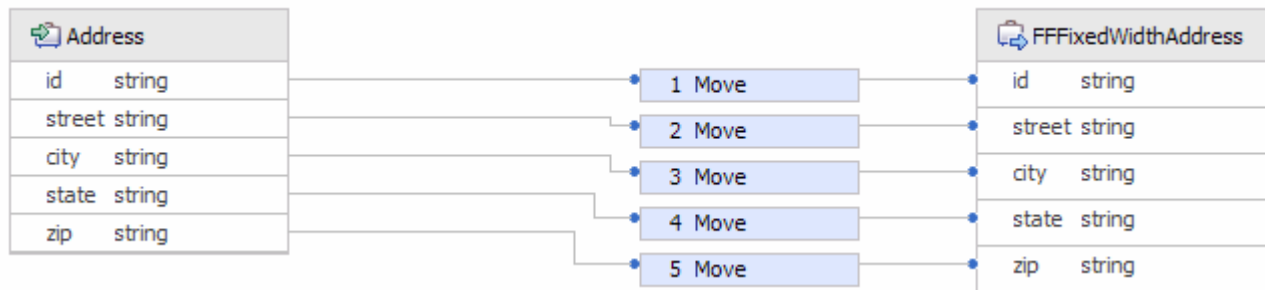


## 4.2. Create data and interface maps

- \_\_\_ 1. Create AddressToFFixedWidthAddress map
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
  - \_\_\_ c. From the Business Object Map screen, enter **AddressToFFixedWidthAddress** for **Name** and click **Next**
  - \_\_\_ d. Select the following from the Business Object Map screen
    - 1) Under **Inputs**, select the check box for the **Address**
    - 2) Under **Outputs** select the check box for the **FFixedWidthAddress** section
    - 3) Click **Finish**

The AddressToFFixedWidthAddress map will be opened for transformation. Ignore the error that you will see at this point as these would be resolved in the following step.

- \_\_\_ e. Add the following connections



- \_\_\_ f. Save (**Ctrl +S**) your changes and close the AddressToFFixedWidthAddress map

---

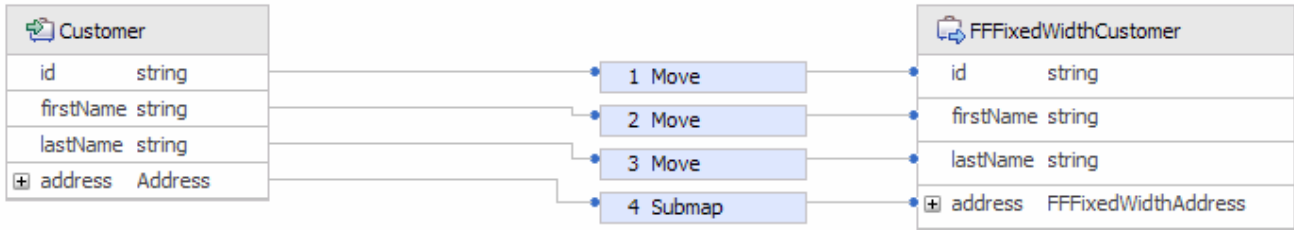
**Note:** You should not see any errors at this point. Ignore any exceptions.

---

- \_\_\_ 2. Create CustomerToFFixedWidthCustomer map
  - \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**
  - \_\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**
  - \_\_\_ c. Enter **CustomerToFFixedWidthCustomer** for the **Name** field and click **Next**
  - \_\_\_ d. From the Business Object Map screen, select the following
    - 1) Under **Inputs**, select the check box for the **Customer**
    - 2) Under **Outputs**, select the check box for **FFixedWidthCustomer**
    - 3) Click **Finish**

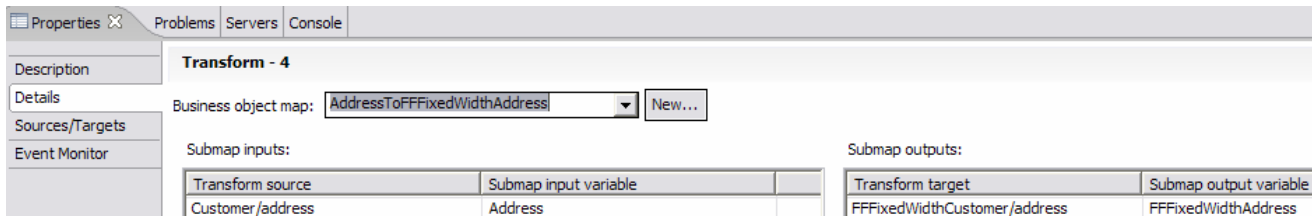
The CustomerToFFFixedWidthCustomer map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

\_\_ e. Add the following connections



\_\_ f. Click on **4 Submap** and then click on **Details** under the **Properties** view from the bottom window

\_\_ g. Select **AddressToFFFixedWidthAddress** for the **Business object map** from the drop down menu



\_\_ h. Save (**Ctrl+S**) your changes and close the CustomerToFFFixedWidthCustomer map

---

**Note:** You should not see any errors at this point. Ignore any exceptions.

---

\_\_\_ 3. Create CustomerToFFFixedWidthCustomerWrapperBG map

\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**

\_\_ b. Right-click on **Data Maps** and select **New > Business Object Map**

\_\_ c. Enter **CustomerToFFFixedWidthCustomerWrapperBG** for the **Name** field and click **Next**

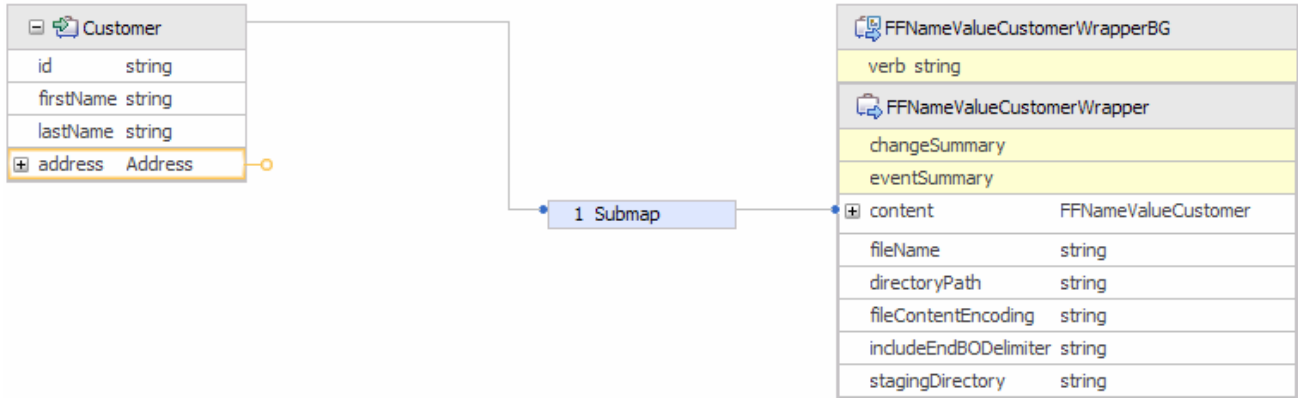
\_\_ d. From the Business Object Map screen, select the following

- 1) Under Inputs, select the check box for the **Customer**
- 2) Under Outputs, select the check box for the **FFFixedWidthCustomerWrapperBG**
- 3) Click **Finish**

The CustomerToFFFixedWidthCustomerWrapperBG map will be opened for transformation. Ignore any errors that you will see at this point. They will be resolved in the following step.

\_\_ e. Add the following connection

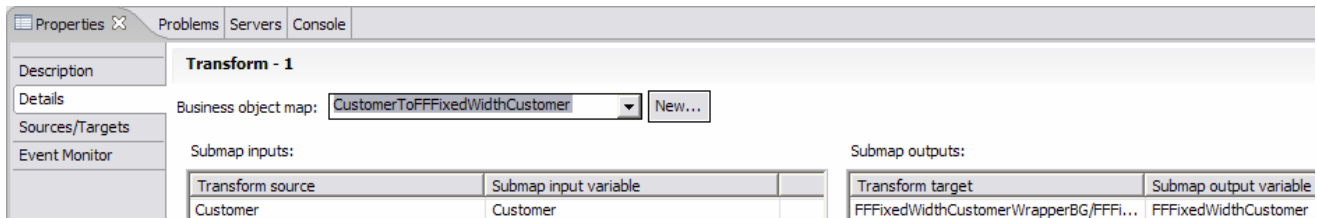
IBM WebSphere Adapter 6.0.2 – Lab Exercise



\_\_\_ f. Click on **1 Submap**

\_\_\_ g. On the bottom window, under Properties view, click on **Details**

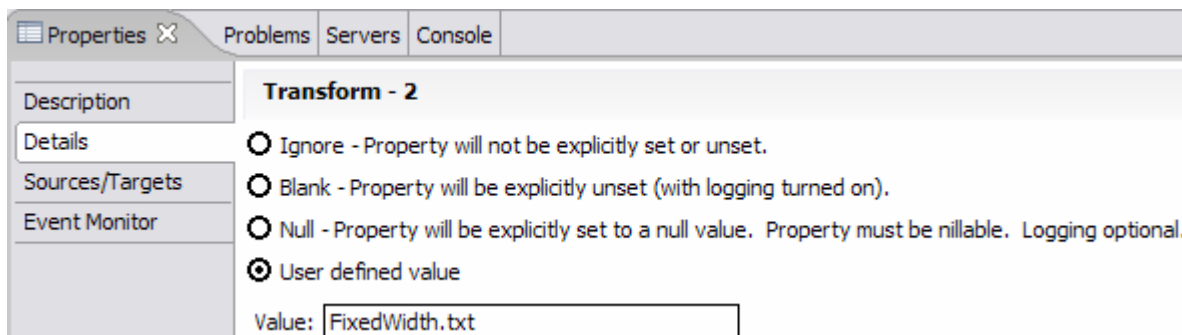
\_\_\_ h. Select **CustomerToFFFixedWidthCustomer** for the **Business object map** from the drop down menu



\_\_\_ i. On the transformations (top window) right-click on **fileName** and select **Create Transform > Assign** from the context menu

\_\_\_ j. Ensure that **Assign** is selected and then click on **Details** under the **Properties** view from the bottom window

\_\_\_ k. Enter **FixedWidth.txt** for Value



\_\_\_ l. Save (**Ctrl+S**) your changes and close the CustomerToFFFixedWidthCustomerWrapperBG map

---

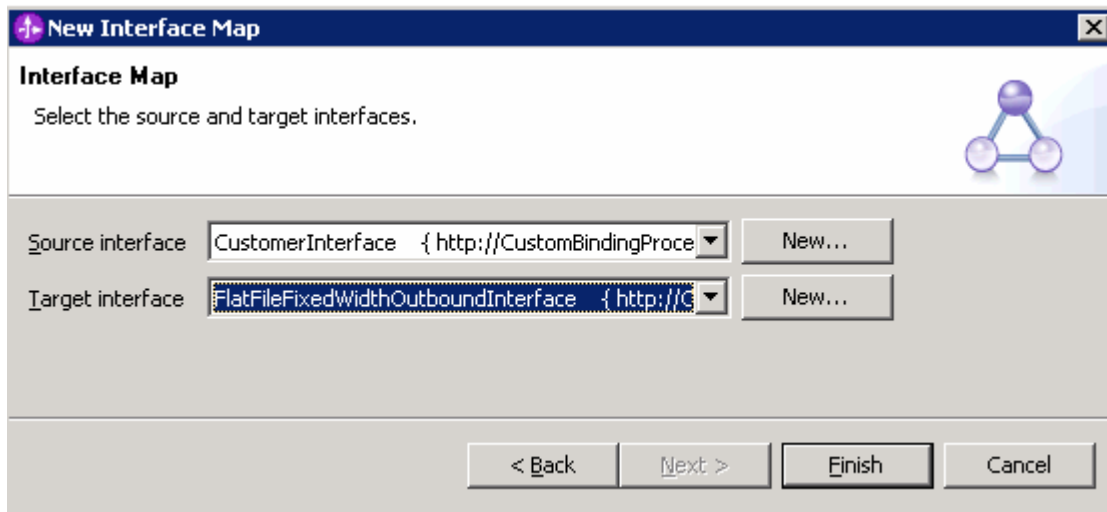
**Note:** You should not see any errors at this point. Ignore any exceptions.

---

\_\_\_ 4. Create CustomerToFixedWidthCustomer Interface Map

\_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing > Mapping**

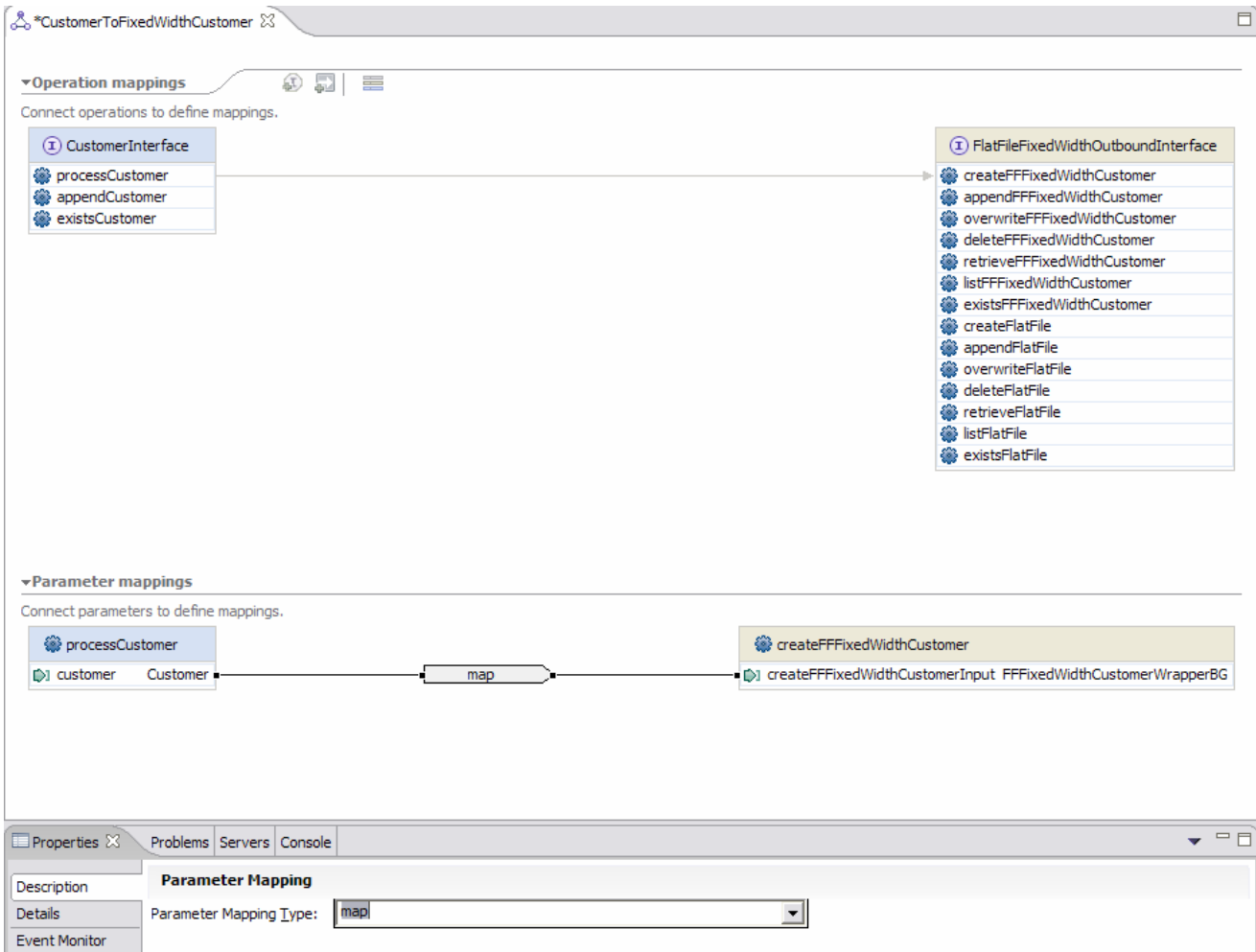
- \_\_\_ b. Right-click on **Interface Maps** and select **New > Interface Map**
- \_\_\_ c. Enter **CustomerToFixedWidthCustomer** for the **Name** and click **Next**
- \_\_\_ d. For **Source interface** field, select **CustomerInterface**
- \_\_\_ e. For **Target interface** select **FlatFileFixedWidthOutboundInterface**
- \_\_\_ f. Click **Finish**



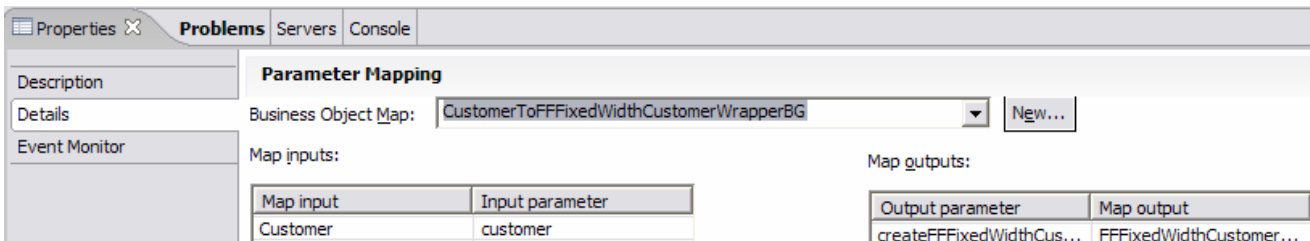
CustomerToFixedWidthCustomer Interface map will be opened for mapping. Ignore any errors at this point. They will be resolved in the following step.

- \_\_\_ g. Create a map from processCustomer to createFFFixedWidthCustomer
  - 1) Click on **processCustomer**, select **Create operation mapping** and then click on **createFFFixedWidthCustomer**
  - 2) Click on the above mapping line and you will see **Parameter mappings** for it
  - 3) Under **Parameter mappings**, create a map from **Customer** to **createFFFixedWidthCustomerInput**
  - 4) Click on **move** and from **Properties** view, select **Description** and then select **map** from the drop down menu for the **Parameter Mapping Type** field

IBM WebSphere Adapter 6.0.2 – Lab Exercise

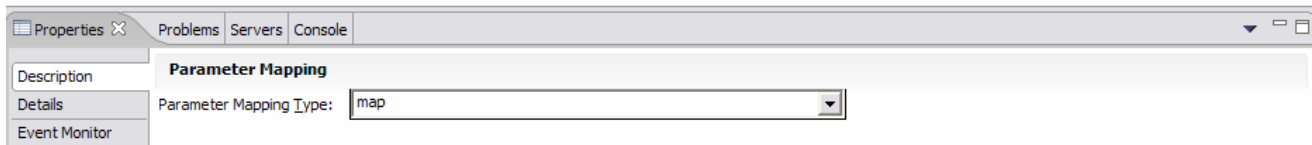
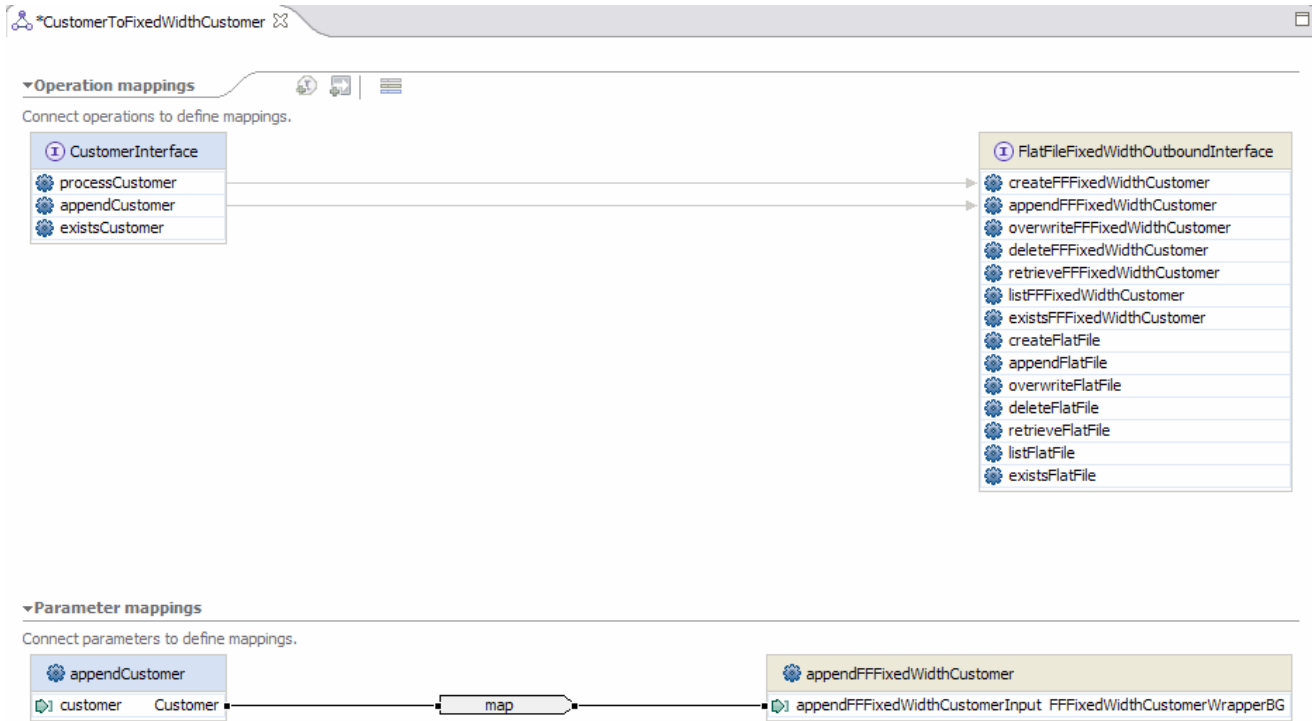


- 5) From **Properties** view, select **Details** tab then select **CustomerToFFixedWidthCustomerWrapperBG** from the drop down menu for the **Business Object Map** field

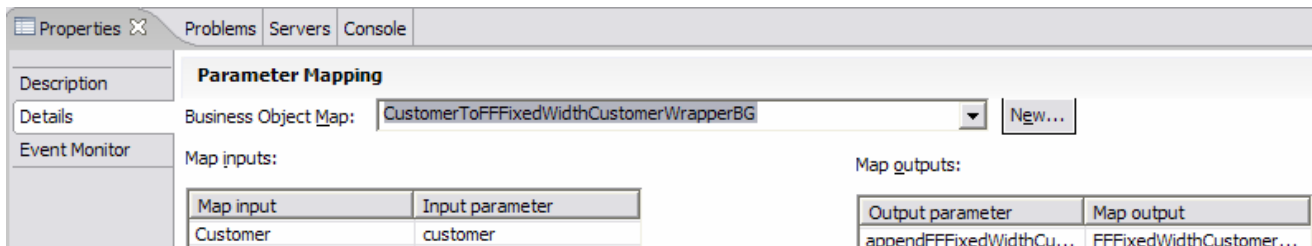


- \_\_\_ h. On the operation mappings, create a map from **appendCustomer** to **appendFFixedWidthCustomer**
  - 1) Click on **appendCustomer**, select **Create mapping** and then click on **appendFFixedWidthCustomer**
  - 2) Click on the above mapping line and you will see **Parameter mappings** for it
  - 3) Under **Parameter mappings**, create a map from **Customer** to **appendFFixedWidthCustomerInput**

- 4) Click on **move** and from **Properties** view, select **Description** and then select **map** from the drop down menu for the **Parameter Mapping Type** field



- 5) From **Properties** view, select **Details** tab and then select **CustomerToFFixedWidthCustomerWrapperBG** from the drop down menu for the **Business Object Map** field

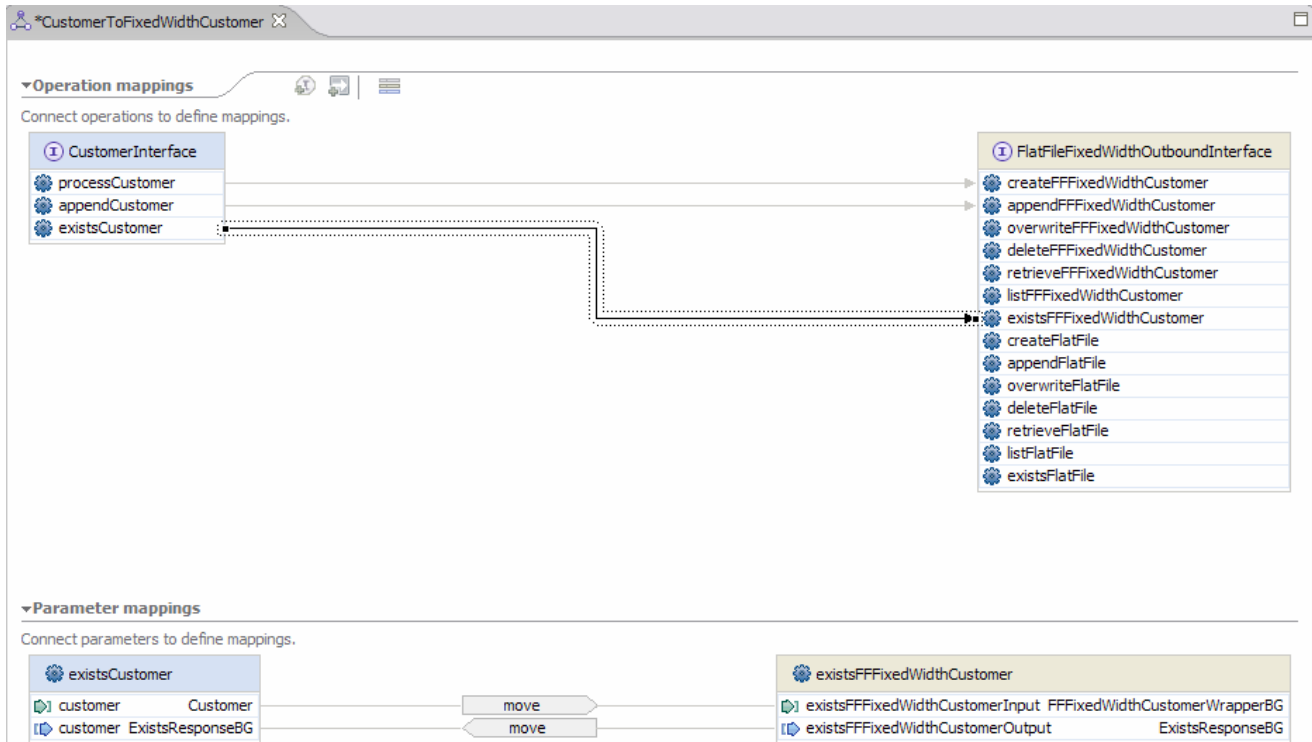


\_\_\_ i. On the operation mappings create a map from existsCustomer to existsFFixedWidthCustomer

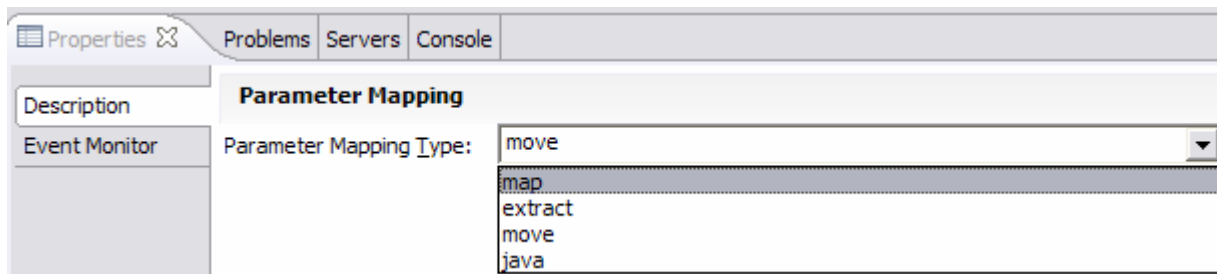
- 1) Click on **existsCustomer**, select **Create operation mapping** and then click on **existsFFixedWidthCustomer**
- 2) Click on the above mapping line to see **Parameter mappings** for it
- 3) Create the following mappings under **Parameter mappings**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

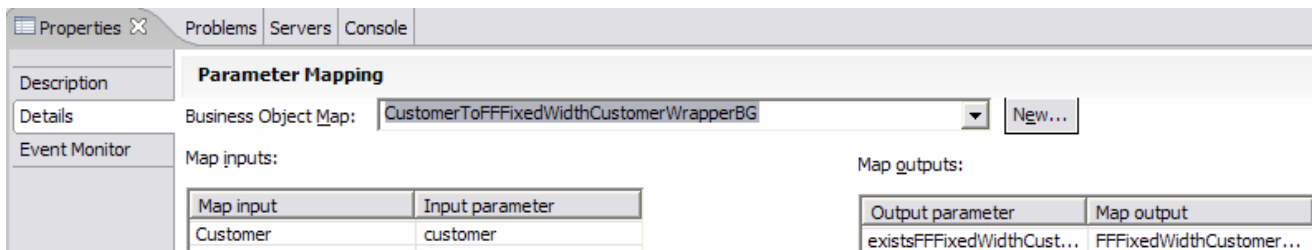
- a) from **Customer** to **existsFFixedWidthCustomerInput**
- b) from **existsFFixedWidthCustomerOutput** to **ExistsResponseBG**



- 4) Click on **move**, connecting from Customer to existsFFixedWidthCustomerInput, and from **Properties** view, select **Description** and then select **map** from the drop down menu for the **Parameter Mapping Type** field



- 5) From **Properties** view, select **Details** tab and then select **CustomerToFFixedWidthCustomerWrapperBG** from the drop down menu for the **Business Object Map** field



- \_\_\_ j. Save (**Ctrl+S**) your changes and close the CustomerToFixedWidthCustomer interface map window

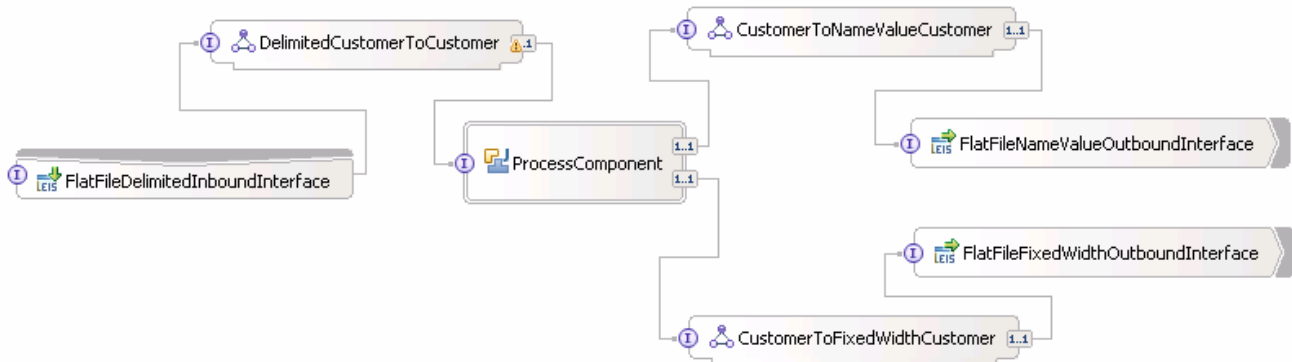
---

**Note:** You should not see any errors at this point. Ignore any exceptions

---

- \_\_\_ 5. Wire the components
- \_\_\_ a. From the Business Integration window, expand **CustomBindingProcessing** and double-click on **CustomBindingProcessing** to open it in Assembly Editor
  - \_\_\_ b. Select **CustomBindingProcessing > Mapping > Interface Maps > CustomerToFixedWidthCustomer** and drop it in the Assembly diagram
  - \_\_\_ c. Add a wire from **ProcessComponent** to **CustomerToFixedWidthCustomer**. Click **OK** and **Yes** for the two pop-up windows.
  - \_\_\_ d. Add a wire from **CustomerToFixedWidthCustomer** to **FlatFileFixedWidthOutboundInterface**

You will see the following in the Assembly diagram



- \_\_\_ 6. Save (**Ctrl+S**) your changes on Assembly diagram
- \_\_\_ 7. Regenerate Implementation for ProcessComponent
- \_\_\_ a. Right-click on **ProcessComponent** and select **Regenerate Implementation** from the context menu
  - \_\_\_ b. The **ProcessComponentImpl.java** will be opened for you. You can see the added CustomerInterface method.
  - \_\_\_ c. Close the **ProcessComponentImpl.java** window
- \_\_\_ 8. Save (**Ctrl+S**) your changes and close the Assembly diagram



---

## 4.3. Run intermediate test

---

**Note:** Refer to the **Appendix: Troubleshooting Tips** at the end of this lab for any exceptions/errors that you might see while running the intermediate test.

---

- \_\_\_ 1. Start WebSphere Process Server test environment and add the **CustomBindingProcessingApp** as shown in **Part 2.4 Run Intermediate Test**

Wait for the project to be added to the server and the application to start. The server will be started in Debug mode if it is not already started before.

- \_\_\_ 2. Test the adapter application
- \_\_\_ a. From the Business Integration perspective, expand **CustomBindingProcessing** and double-click on **Assembly Diagram** to open it in assembly editor
  - \_\_\_ b. right-click on the **FlatFileFixedWidthOutboundInterface** and select **Test Component**

The FlatFileFixedWidthOutboundInterface\_Test will be opened

- \_\_\_ c. Under **Detailed Properties**, for the **Operation** field, select **createFFFixedWidthCustomer** from the drop down menu
- \_\_\_ d. Fill out the fields for Initial request parameters:
  - 1) For content and address fields enter some data of your choice similar to what is shown on the image
  - 2) fileName: **FixedWidth.txt**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

**▼ Detailed Properties**

**Configuration:** Default Module Test

**Module:** CustomBindingProcessing

**Component:** FlatFileFixedWidthOutboundInterface

**Interface:** FlatFileFixedWidthOutboundInterface

**Operation:** createFFFixedWidthCustomer

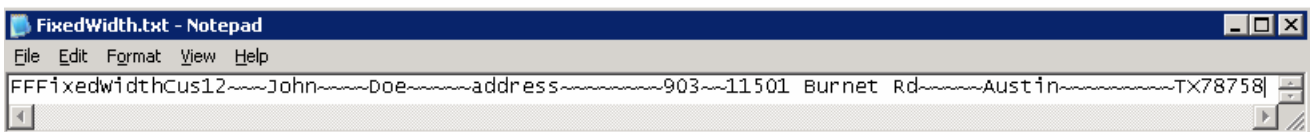
Initial request parameters

Name	Type	Value
[-] createFFFixedWidthCus...	FFFixedWidthCustomerW...	
verb	string	<null>
[-] FFFixedWidthCustom...	FFFixedWidthCustomerW...	
[-] content	FFFixedWidthCustomer	
id	string	123
firstName	string	John
lastName	string	Doe
[-] address	FFFixedWidthAddress	
id	string	11501
street	string	Burnet Rd
city	string	Austin
state	string	TX
zip	string	78758
fileName	string	FixedWidth.txt
directoryPath	string	
fileContentEncoding	string	
includeEndBODeli...	string	
stagingDirectory	string	

- \_\_\_ e. Click **Continue**
- \_\_\_ f. From Deployment Location window, select **WebSphere Process Servers > WebSphere Process Server V6.0** and click **Finish**
- \_\_\_ 3. Verify the created file contents
  - \_\_\_ a. Check the subdirectory **<DATA\_DIR>** for the file **FixedWidth.txt**



- \_\_\_ b. Double-click on **FixedWidth.txt** to open it and you should see the contents that you filled in the above steps:



- \_\_\_ 4. Restore the Sever Configuration
- \_\_\_ a. Close **FlatFileFixedWidthOutboundInterface\_Test** window and select **No** from the Save Resource pop up window
  - \_\_\_ b. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
  - \_\_\_ c. Select **CustomBindingProcessingApp** under Configured projects and click **< Remove**
  - \_\_\_ d. Click **Finish** after you see the application moved to Available projects. Wait until the application is being unpublished
  - \_\_\_ e. Delete the **FixedWidth.txt** file from **<DATA\_DIR>**

## Part 5: Test the scenario

In this part of the lab, you will use the WebSphere Process Server Test Environment to test complete scenario of this lab.

- \_\_\_ 1. Generate Implementation for the ProcessComponent
  - \_\_\_ a. Open the assembly diagram for CustomBindingProcessing
  - \_\_\_ b. Right-click on ProcessComponent and select Regenerate Implementation
  - \_\_\_ c. The ProcessComponentImpl.java window will be opened. From this window, scroll down to the **processCustomer** method and enter the following code:

```

public void processCustomer(DataObject customer) {
    DataObject existResponseBG = locateService_CustomerInterfacePartner().existsCustomer(customer);
    DataObject existsResponse = existResponseBG.getDataObject("ExistsResponse");
    boolean doesExists = existsResponse.getBoolean("doesFileExist");

    if(doesExists)
        locateService_CustomerInterfacePartner().appendCustomer(customer);
    else
        locateService_CustomerInterfacePartner().processCustomer(customer);

    DataObject existResponseBG1 = locateService_CustomerInterfacePartner1().existsCustomer(customer);
    DataObject existsResponse1 = existResponseBG1.getDataObject("ExistsResponse");
    boolean doesExists1 = existsResponse1.getBoolean("doesFileExist");

    if(doesExists1)
        locateService_CustomerInterfacePartner1().appendCustomer(customer);
    else
        locateService_CustomerInterfacePartner1().processCustomer(customer);
    System.out.println("*****Inside processCustomer*****");
}

```

---

**Note:** For you convenience, the code is available at <LAB\_FILES>\DTF\ProcessComponentImpl.txt file

---

- \_\_\_ d. Save (**Ctrl+S**) your changes and close the ProcessComponentImpl.java window
- \_\_\_ e. Save (**Ctrl+S**) your changes to the Assembly diagram
- \_\_\_ 2. Start the WebSphere Process Server (if it is not already started)
  - \_\_\_ a. Select the **Servers** view
  - \_\_\_ b. Right-click on the row that contains **WebSphere Process Server V6.0** and select **Start** from the context menu
  - \_\_\_ c. Wait until the server status shows as **Started**
- \_\_\_ 3. Add the project to the WebSphere Test Environment server
  - \_\_\_ a. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
  - \_\_\_ b. From the Add and Remove Projects window, select **CustomBindingProcessingApp** under Available projects panel and click **Add >**
  - \_\_\_ c. You will now see the **CustomBindingProcessingApp** added to the **Configured projects**

IBM WebSphere Adapter 6.0.2 – Lab Exercise

\_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server will be started in Debug mode if it is not already started before

\_\_\_ 4. Test the adapter application

\_\_\_ a. Copy a test file to the **<EVENT\_DIR>**. As the file is copied to the event directory, the same will be picked up and processed by the adapter and will be moved to the **<ARCHIVE\_DIR>** and also the **NameValue.txt** and **FixedWidth.txt** files will be created under **<DATA\_DIR>**

**Note:** For your convenience, a test file named **Delimited.txt** file can be found in **<LAB\_FILES>\DTF**.

\_\_\_ 5. Verify your test results

\_\_\_ a. Inbound operation verification:

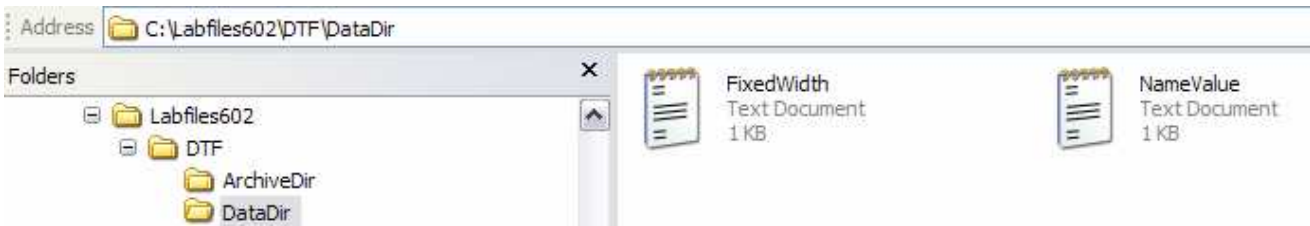
1) Check your test environment console (or SystemOut.log file) for the following successful message:

```
0000008f ResourceAdapt I com.ibm.j2ca.extension.eventmanagement.internal.EventSender sendEvent() CWYBS050SI: The event has been delivered.
000000d5 SystemOut O *****Inside processCustomer*****
```

2) Now, 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.

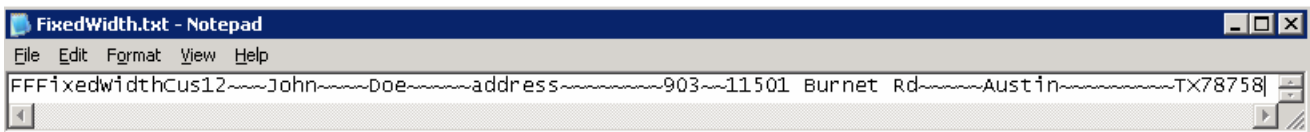
\_\_\_ b. Outbound operation verification:

1) Check for the **NameValue.txt** and **FixedWidth.txt** files created under **<DATA\_DIR>**

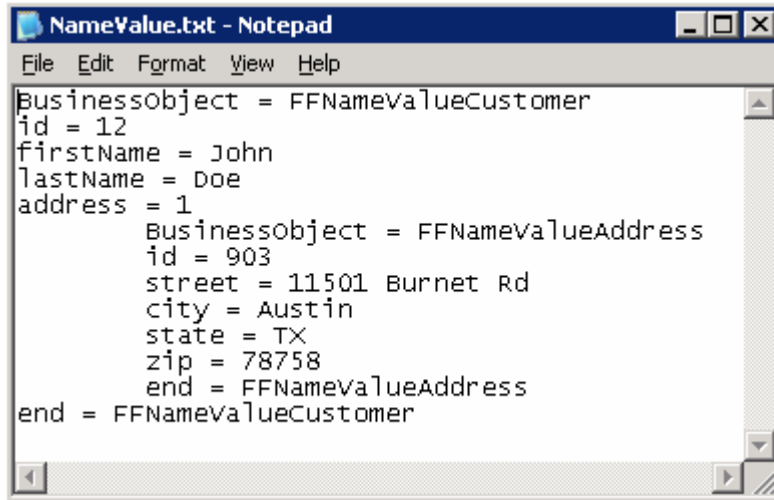


2) You can open the above created files and see the content in it

a) Open **FixedWidth.txt**



b) Open **NameValue.txt**



```
BusinessObject = FFNameValueCustomer
id = 12
firstName = John
lastName = Doe
address = 1
    BusinessObject = FFNameValueAddress
    id = 903
    street = 11501 Burnet Rd
    city = Austin
    state = TX
    zip = 78758
    end = FFNameValueAddress
end = FFNameValueCustomer
```


- \_\_\_ 6. Restore the Sever Configuration
  - \_\_\_ a. Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
  - \_\_\_ b. Select **CustomBindingProcessingApp** 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
  - \_\_\_ d. Stop the server

## Appendix: Troubleshooting tips

This part will help you with some tips when you see some exceptions/errors while doing this lab.

- \_\_\_ 1. Delete the spaces and new line (if added) from the xsds
  - \_\_\_ a. If the new line is added to the target namespace in FlatFile.xsd and FFDelimitedCustomerWrapper.xsd or any other xsds, delete the spaces and new lines from them and save those changes
  - \_\_\_ b. For example, if your TargetNameSpace and DataBindingClass is as shown:
 

```
<TargetNamespace>
    http://www.ibm.com/crm
</TargetNamespace>
<DataBindingClass>
    com.ibm.websphere.samples.databinding.DelimitedEISDataBinding
</DataBindingClass>
```
  - \_\_\_ c. Change it to the following (delete the space and new line):
 

```
<TargetNamespace>http://www.ibm.com/crm</TargetNamespace>
<DataBindingClass>com.ibm.websphere.samples.databinding.DelimitedEISDataBinding</DataBindingClass>
```
- \_\_\_ 2. Clean and Build the Project – If you see any errors regarding the synchronizing problem of WebSphere Integration Developer. For example, after running ESD wizard, you might see a WSDL Port Type error which you can get rid off by doing this step.
  - \_\_\_ a. From the main menu, select **Project > Clean**
  - \_\_\_ b. Select radio button next to **Clean all projects** and click **OK**
- \_\_\_ 3. Restart the server
  - \_\_\_ a. Click on the **Servers** tab
  - \_\_\_ b. Ensure that **WebSphere Process Server V6.0** is selected
  - \_\_\_ c. Click on Restart the server (  ) icon and wait for the server to start
- \_\_\_ 4. Restart workspace – because of the WebSphere Integration Developer V6.0.2 synchronization issues
  - \_\_\_ a. From main menu, select **File > Switch Workspace** and then select **<WORKSPACE>** from the Workspace Launcher window
- \_\_\_ 5. See the server logs
  - \_\_\_ a. From the windows explorer, select `<WID_HOME>\pf\wps\logs\server1`
  - \_\_\_ b. Check the contents in SystemOut.log and SystemErr.log mainly
- \_\_\_ 6. Republish the application – check server log files and if there is anything that you missed in ESD wizard, or entered incorrect information, then make those corrections and then do republish of the application
  - \_\_\_ a. Remove application

*IBM WebSphere Adapter 6.0.2 – Lab Exercise*

- 1) Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
- 2) From the Add and Remove Projects window, select **CustomBindingProcessingApp** under Configured projects panel and click **Remove >**
- 3) You will now see the **CustomBindingProcessingApp** added to the **Available projects**
- 4) Click **Finish**

\_\_\_ b. Add the application

- 1) Right-click on **WebSphere Process Server V6.0** under the Servers view and select **Add and remove projects...** from the context menu
- 2) From the Add and Remove Projects window, select **CustomBindingProcessingApp** under Available projects panel and click **Add >**
- 3) You will now see the **CustomBindingProcessingApp** added to the **Configured projects** and click **Finish**



## What you did in this exercise

In this lab, you used ESD to configure the Flat File Adapter for Inbound operation to process Delimited files and tested it independently with a Delimited test file. Then, you continued using ESD to configure Flat File Adapter for two outbound operations to process NameValue and FixedWidth files and tested these two output configurations independently.

At the end, you tested the entire scenario, where the Delimited file is polled from the event directory and then the two – NameValue and FixedWidth files were sent to the output directory as configured using only one Module.

This page is left intentionally blank.