FTP Inbound Lab

What this exercise is about

The objective of this lab is to provide you with an understanding of the WebSphere® Adapter for FTP and inbound event processing. In this lab you will install and deploy the WebSphere Adapter for FTP and create an SCA application that polls for and processes inbound events from the file system.

Lab Requirements

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

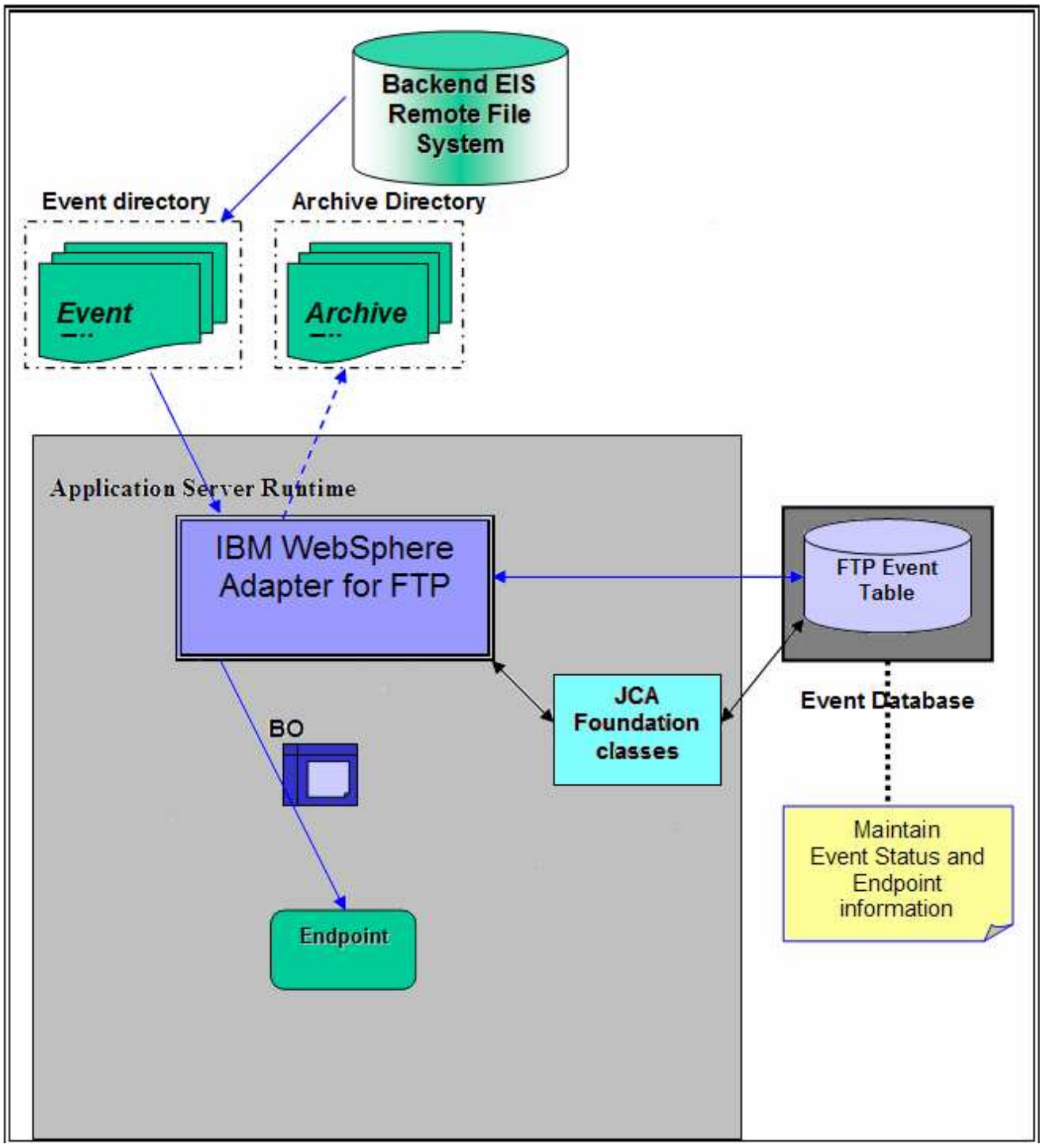
- WebSphere Integration Developer V6.0.2 installed
- WebSphere Process Server V6.0 Test Environment installed
- WebSphere Adapter for FTP V6.0.2 installed
- Unzip LabFiles602.zip to your C:\ (your root) drive
- FTP server installed and configured

What you should be able to do

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

- Import FTP adapter RAR file into WebSphere Integration Developer
- Use WebSphere Process Server's Administrative Console to create required JDBC provider and a Data source
- Use Enterprise Service Discovery (ESD) wizard to configure the Activation Spec Properties and the Resource Adapter Properties to generate Business Objects and other artifacts
- Deploy the adapter application onto the WebSphere Process Server test environment
- Test the deployed application using WebSphere Process Server test environment for both pass-through and non pass-through using four different scenarios
- Restore the server configuration

Introduction



The backend EIS is the source of events. When events are generated at the EIS, files will be created by the EIS in the remote file system at a specific directory location. The same directory needs to be configured by the user as the Event Directory for the adapter.

The adapter polls event files, based on configured Event File Mask and FTP Get Quantity, from the Event Directory and downloads to a Local Event Directory in the adapter machine for every FTP Poll Frequency time and archives the file in FTP Archive Directory of the FTP server (it specified by the user).

The adapter splits the downloaded event files from Local Event Directory based on the configured SplittingFunctionClassname (splitting functionality) and SplitCriteria (splitting criteria used). User can implement his/her own class which contains the splitting logic which splits the event files into individual Business Objects. The adapter provides a Java™ interface which the end user has to implement in a class.

An entry is made in the Event Persistence table (given by EPEventTableName) for each BO with status as New Event. The adapter sends the record chunk, through a Function Selector and Data Binding, to the endpoint and the status in the Event Persistence table is updated based on successful (to PROCESSED state) or failed(to FAILED state) delivery to the endpoint. The event management framework takes care of delivering the event only once to the endpoint.

If archiving is enabled (if LocalArchiveDirectory has a valid value), the Business Objects are also archived in a configured Local Archive Directory, on the local file system, based on successful or failed delivery to the endpoint. Successful and failed events are written to different files. The entries from the Event Persistence table are deleted for each of the successfully processed BO's only. If Local ArchiveDirectory is not set, the event file is deleted after processing of all the BO's is completed.

If EventContentType in the Activation Spec Properties is not set or if it is not valid or the value does not match the entries present in annotation of the Wrapper data object then it is called PassThrough and data transformation will not happen in this case. During PassThrough the SplittingFunctionClassName and SplitCriteria are set to values (even if they are set to different values) such that splitting happens based on file size. The PassThrough scenario can have either Chunking or FilePassbyReference features.

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	
<FTPADAPTER_HOME>	C:\Program Files\IBM\ResourceAdapters\FTP\adapter\FTP	
<LAB_FILES>	C:\Labfiles602	/tmp/Labfiles602
<WORKSPACE>	<LAB_FILES>\FTPInbound\workspace	
<LOCAL_EVENT_DIR>	<LAB_FILES>\FTPInbound\LocalEventDir	
<LOCAL_ARCHIVE_DIR>	<LAB_FILES>\FTPInbound\LocalArchivetDir	
<TEMP>	C:\temp	/tmp

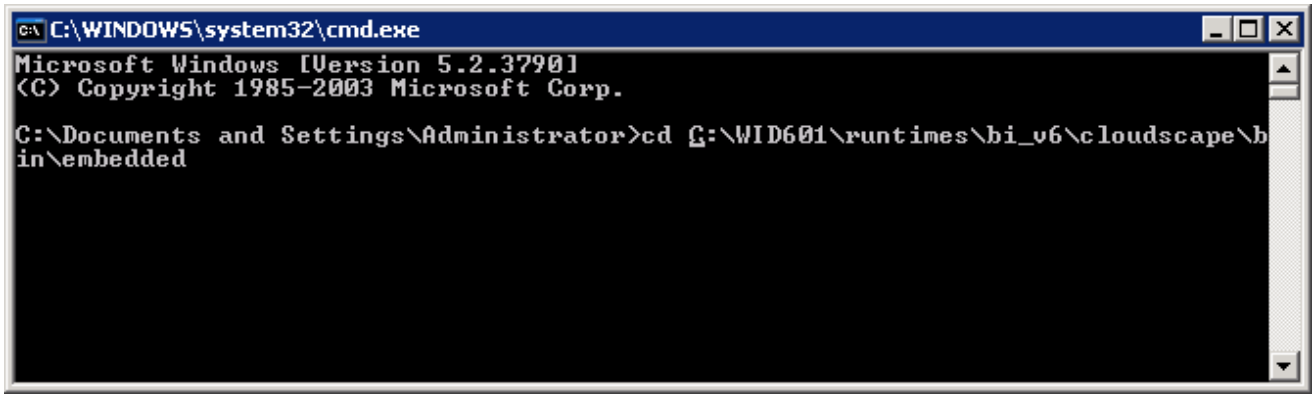
Windows users note: When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, C:\LabFiles602\ would be replaced by C: /LabFiles602/

Part 1: Prepare Database and Directory Structure on your FTP Server

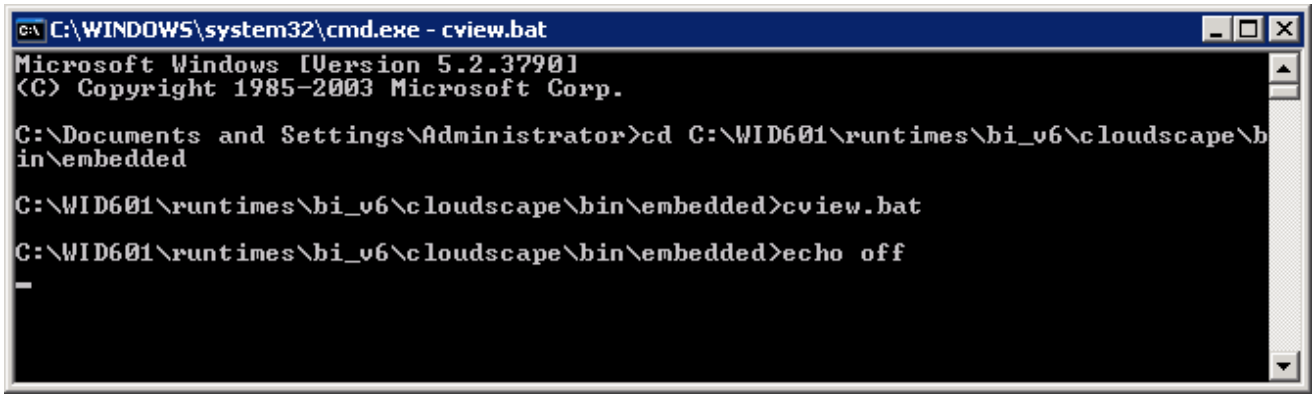
This part of the lab describes the steps for creating the **FTPDATABASE** database in Cloudscape which will contain the Event Distribution Table, **FTPTABLE**. The **FTPTABLE** will be created automatically for you during the install and deployment of the application containing the FTP adapter to the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

____ 1. Start the Cloudscape Cview Graphical User Interface (GUI) by executing the **cview.bat** file

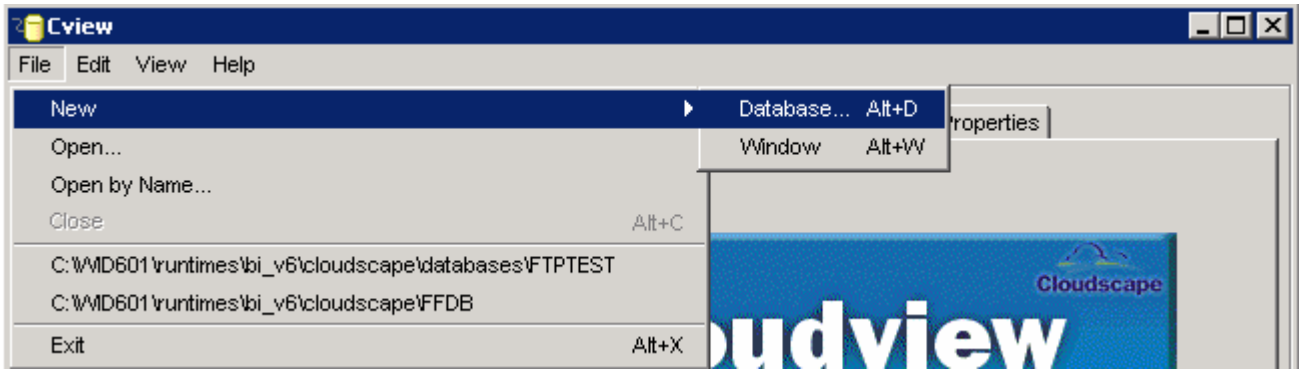
__ a. Open a command prompt window and change to the following subdirectory
<WPS_HOME>\cloudscape\bin\embedded



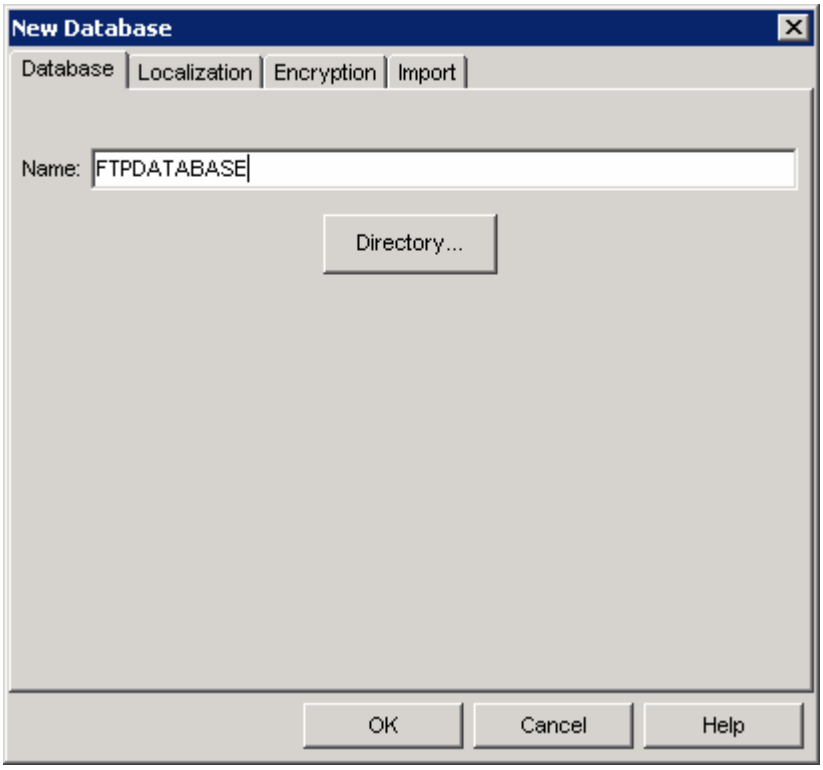
__ b. Type **cview.bat**



__ c. The Cview window will be opened. From that window, select **File > New > Database…**.



__ d. Enter **FTPDATABASE** in the Name field and click **OK**



__ e. You will see the **FTPDATABASE** created in the left pane of the Cview window. Now select **File > Exit** to close the Cview GUI

____ 2. Create directory structure on your FTP Server

__ a. Log onto FTP machine/FTP Server using your ftpuser and password

__ b. Create an **EventDir** and an **ArchiveDir** under the user's home directory:

1) mkdir EventDir

2) mkdir ArchiveDir

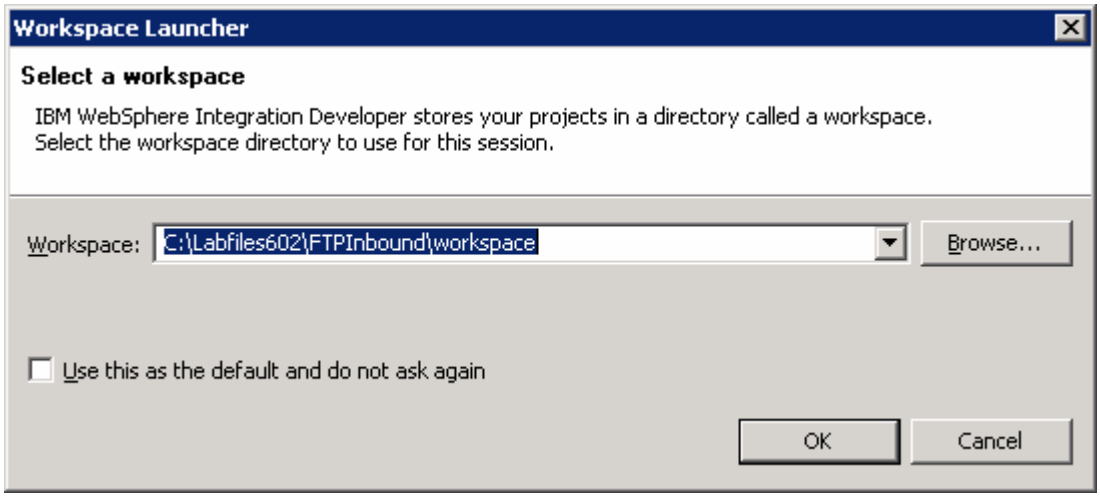
Part 2: Initialize workspace and Import RAR into WebSphere Integration Developer

This part of the lab will guide you through the steps for starting WebSphere Integration Developer with a new workspace, and then import the connector file **CWYFT_FTPFile.rar** into your new workspace.

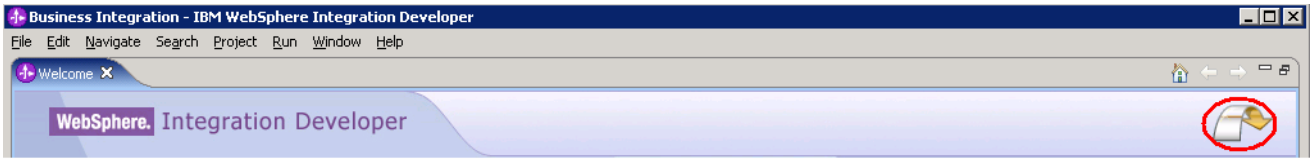
____ 1. Start the WebSphere Integration Developer V6.0.2 with a new workspace

__ a. Select **Start > Programs > IBM WebSphere > Integration Developer V6.0.2 > WebSphere Integration Developer V6.0.2**

__ b. From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field

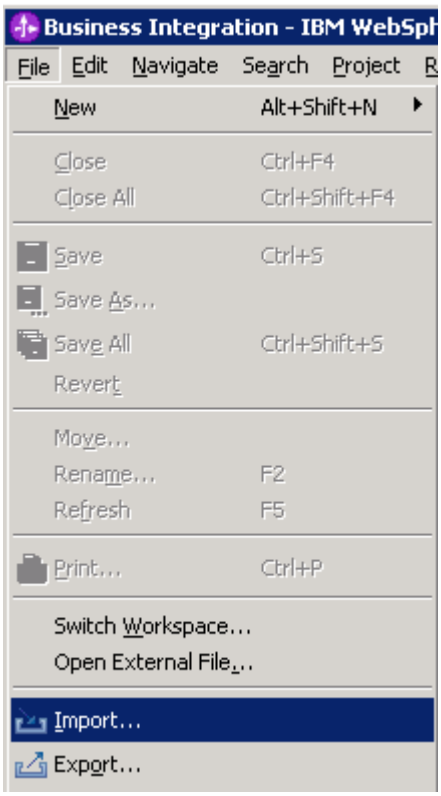


__ c. Click on the button on the right hand corner to close the Welcome page and proceed with the workbench

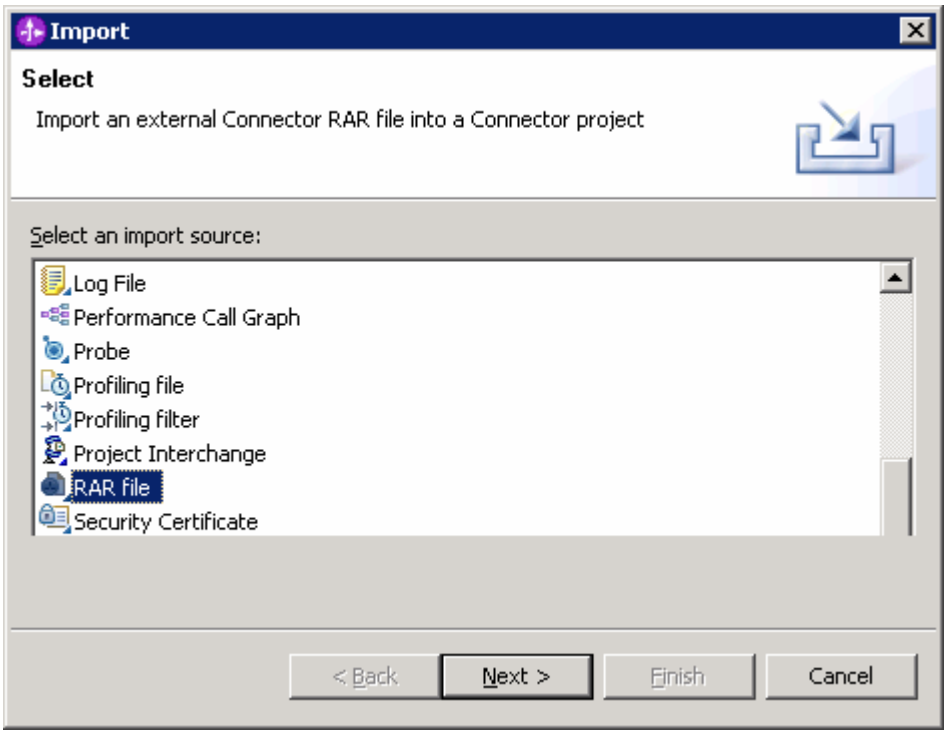


____ 2. Import FTP Adapter RAR file

__ a. From main menu, select **File > Import…**

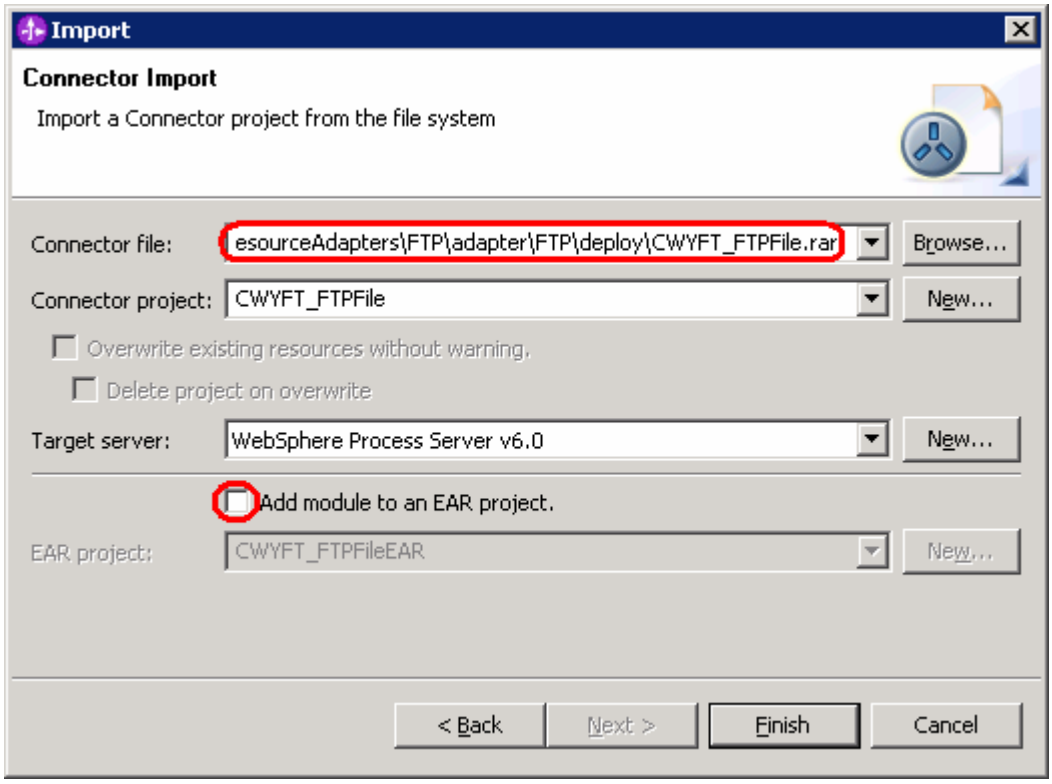


__ b. Select **RAR file** from the Import window and then click **Next**

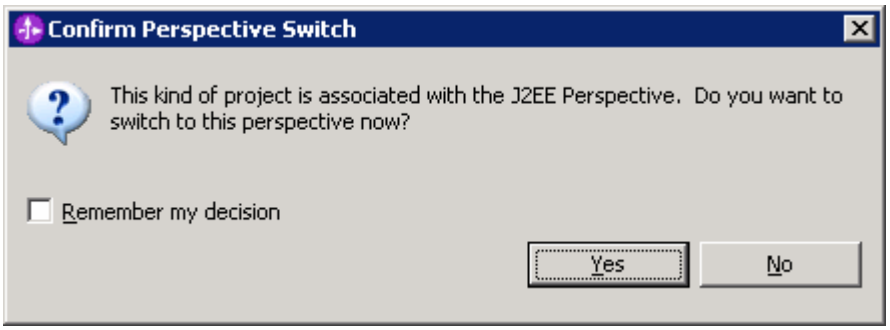


__ c. Click on the **Browse…** button next to the Connector file filed to select **WYFT_FTPFile.rar**

__ d. Uncheck the check box next to **Add module to and EAR project** and click **Finish**



__ e. Click on **No** from Confirm Perspective Switch window to continue with the Business Integration perspective



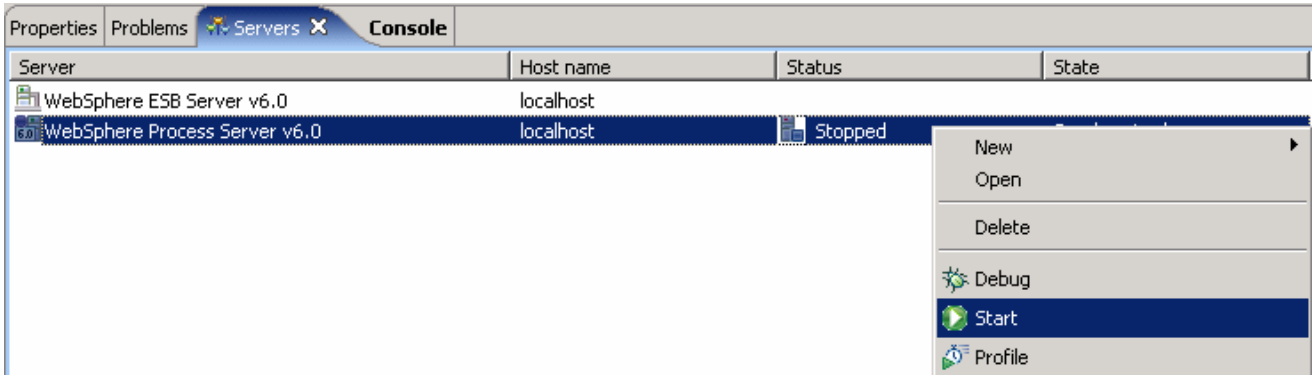
Part 3: Use WebSphere Process Server Administrative Console to Configure Data Sources

In this part of the lab, you will make use of the 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 test Environment

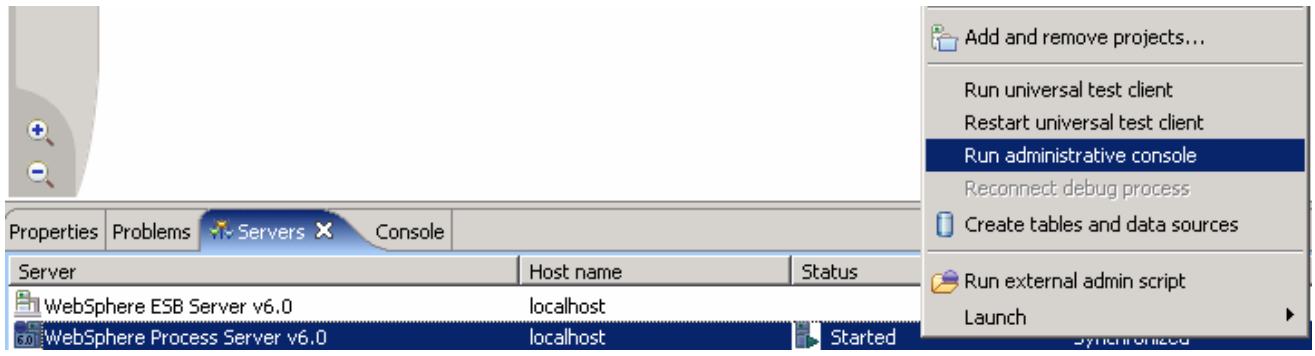
__ a. Select **Servers** view

__ b. Right-click on the row that contains WebSphere Process Server v6.0 and select **Start** from the context menu

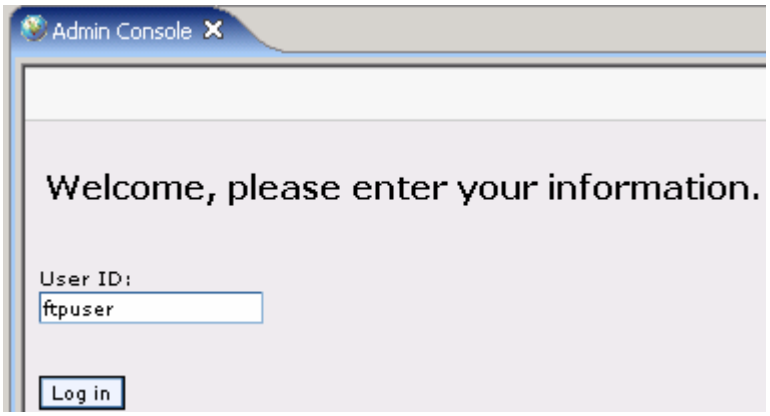


__ c. Wait until the Status of the server shows **Started**

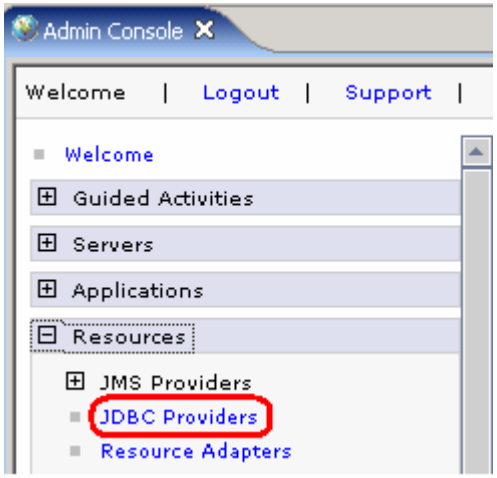
____ 2. Right-click on the row that contains WebSphere Process Server v6.0 and select **Run administrative console** from the context menu



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



____ 4. Expand **Resources** on the left pane and select **JDBC Providers**



____ 5. Accept the default scope and click on **New**

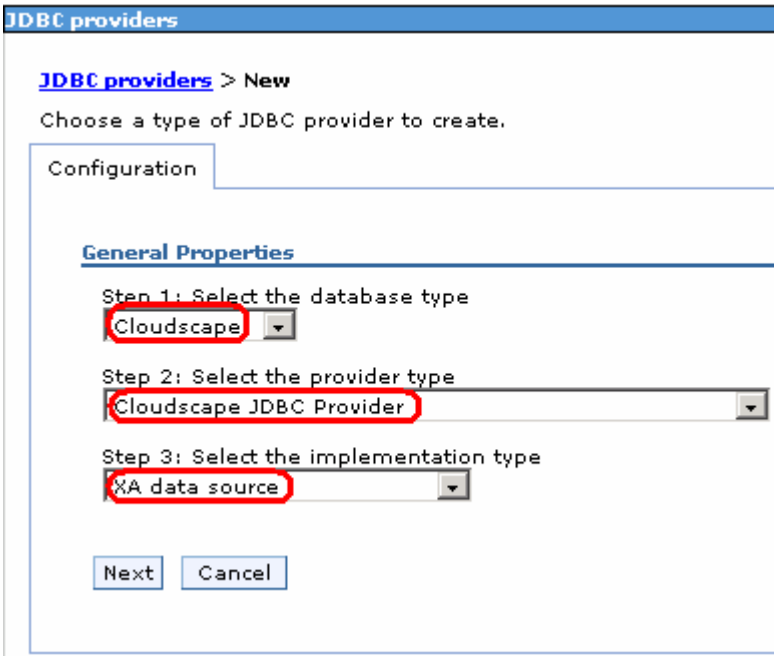
____ 6. In the following screen, for the General Properties, select the following from the dropdown list:

__ a. Step 1: **Cloudscape**

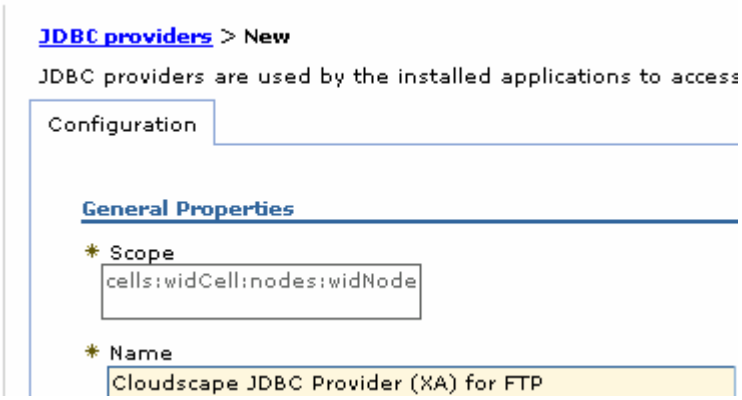
__ b. Step 2: **Cloudscape JDBC Provider**

__ c. Step 3: **XA data source**

____ 7. Click **Next**



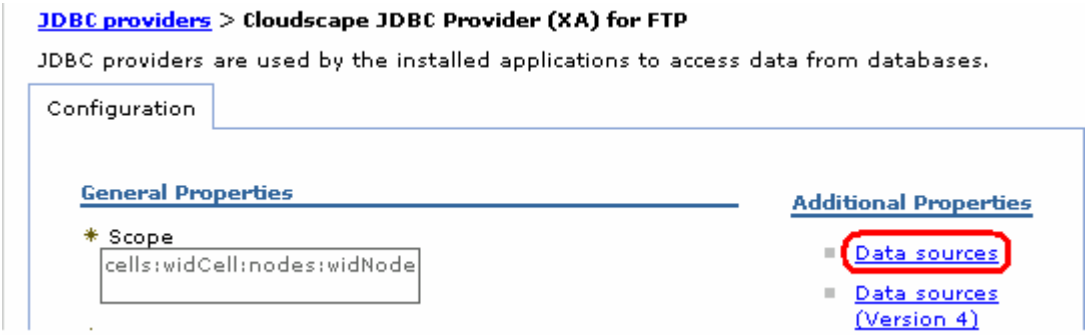
____ 8. Enter **Cloudscape JDBC Provider (XA) for FTP** in the **Name** field and then click **OK**



____ 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 FTP** 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**

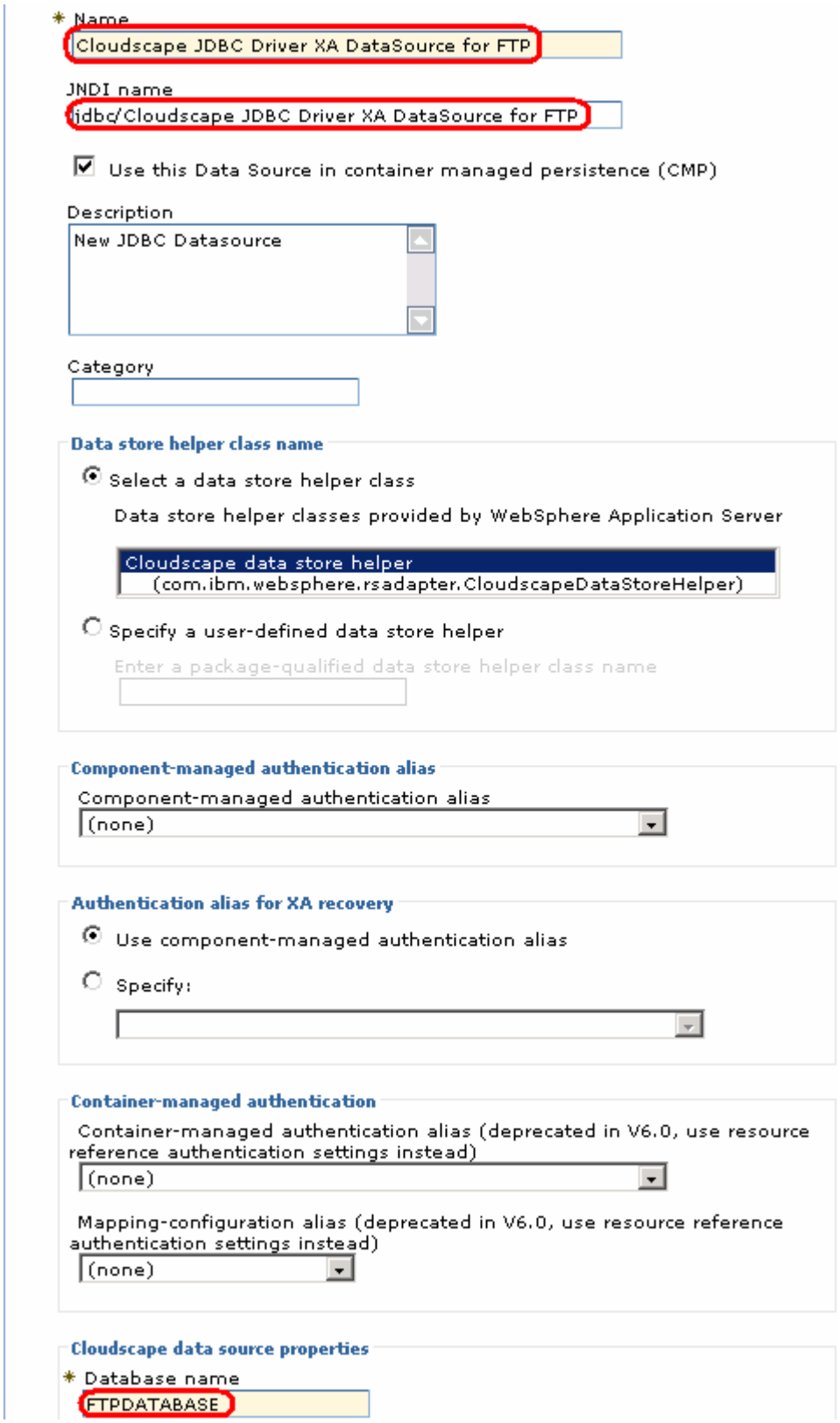
__ b. Enter the following:

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

b) JNDI Name: **jdbc/Cloudscape JDBC Driver XA DataSource for FTP**

c) Database name: **FTPDATABASE**

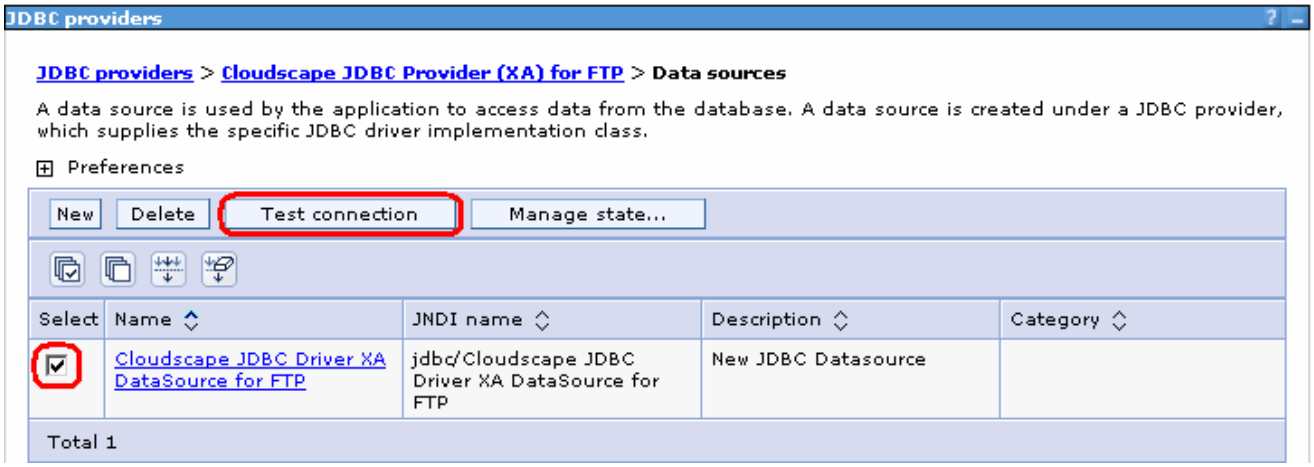
__ c. Click **OK**



____ 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 FTP** and click on **Test connection** from the top of the screen

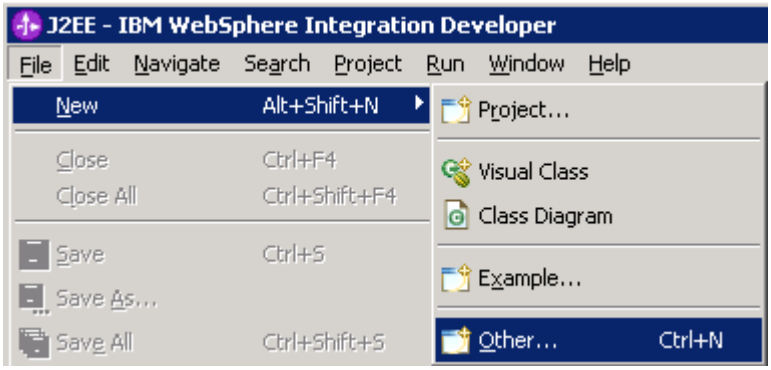


__ b. You should see the following success message on the top of the screen:

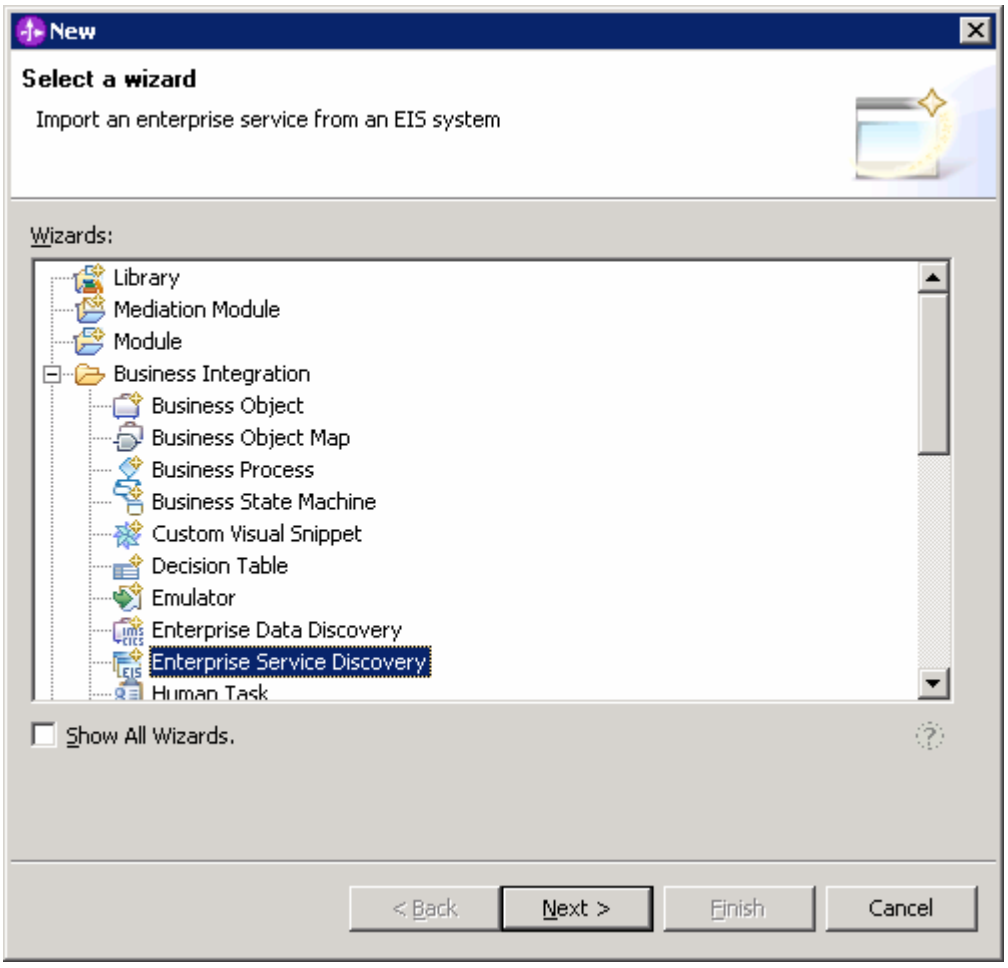
Part 4: Use Enterprise Service Discovery Wizard to Generate Business Objects and other Artifacts

____ 1. Launch the Enterprise Service Discovery wizard

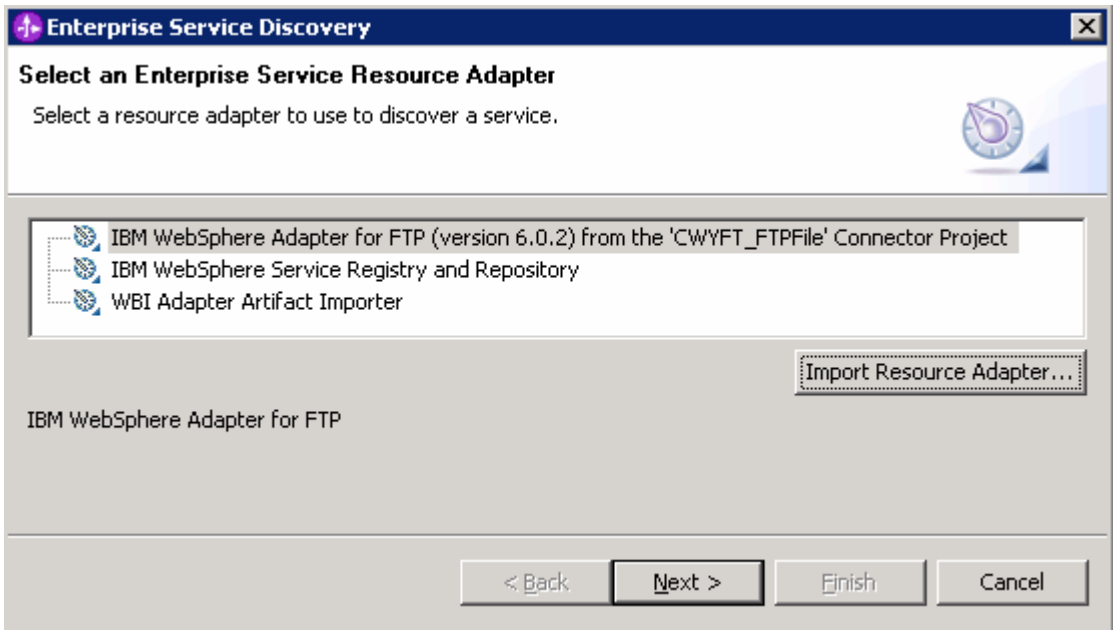
__ a. Select **File > New > Other…**.



__ b. From the New window, select **Business Integration > Enterprise Service Discovery** and click **Next**



____ 2. Select **IBM WebSphere Adapter for FTP (version 6.0.1) from the 'CWYFT_FTPFile' Connector Project** and click **Next**



____ 3. Configure settings for the Discovery agent

You will specify the properties to initialize the Resource Adapter and Enterprise Service Discovery agent

__ a. Click on the **Browse…** button next to the **Folder Name** field and select the folder **<FTPADAPTER_HOME>\samples** that contains the XSD file for **Customer** Business Object

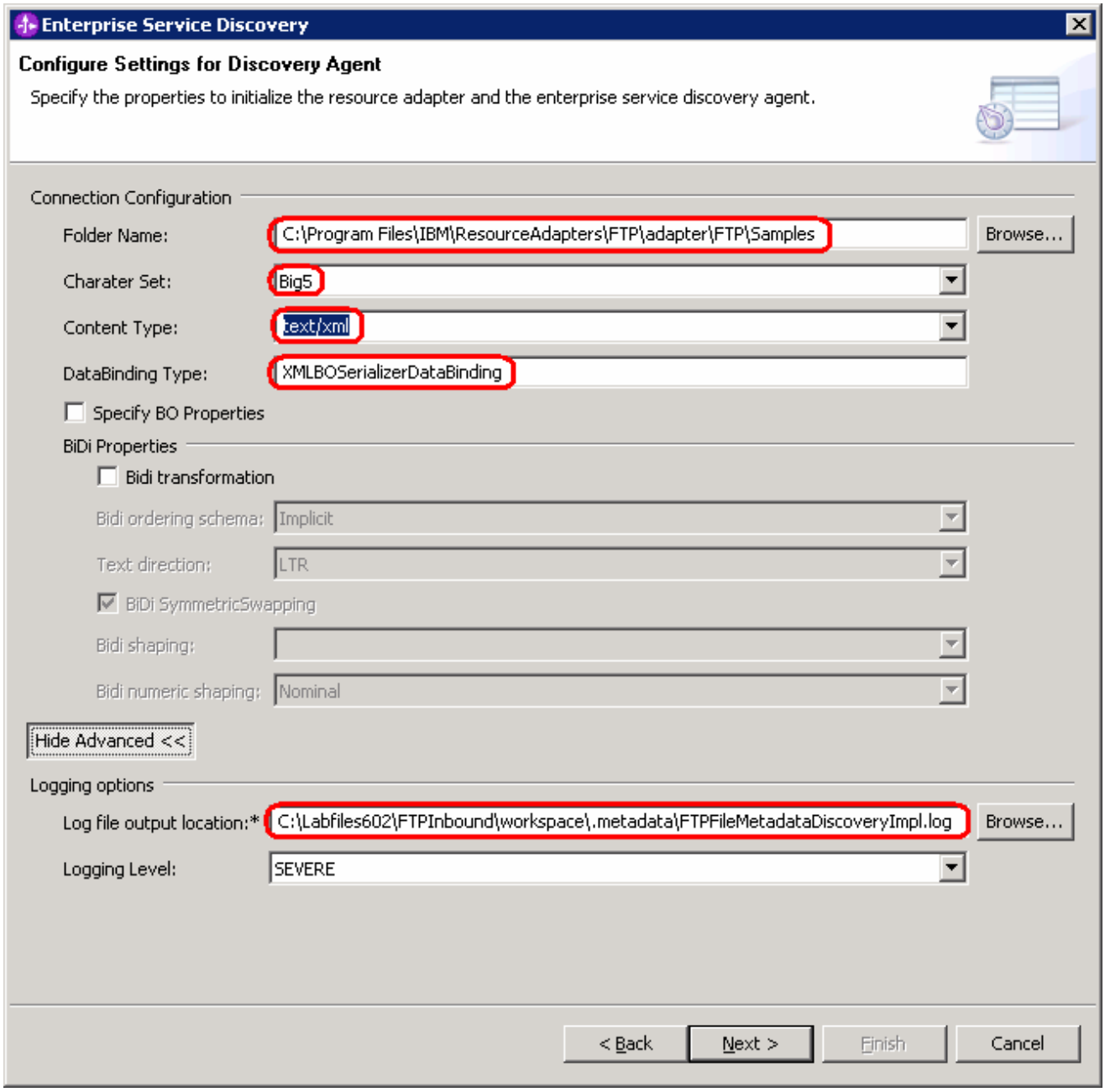
---

Note: For your convenience, the **Customer.xsd** is also placed under **<LAB_FILES>\FTPFiles**.

__ b. Select **Big5** from the drop down list for the **Character Set** field

__ c. Select **text/xml** for the **Content Type** field. Once the content type is selected, the **DataBinding Type** field will be automatically set to **XMLBOSerializerDataBinding**

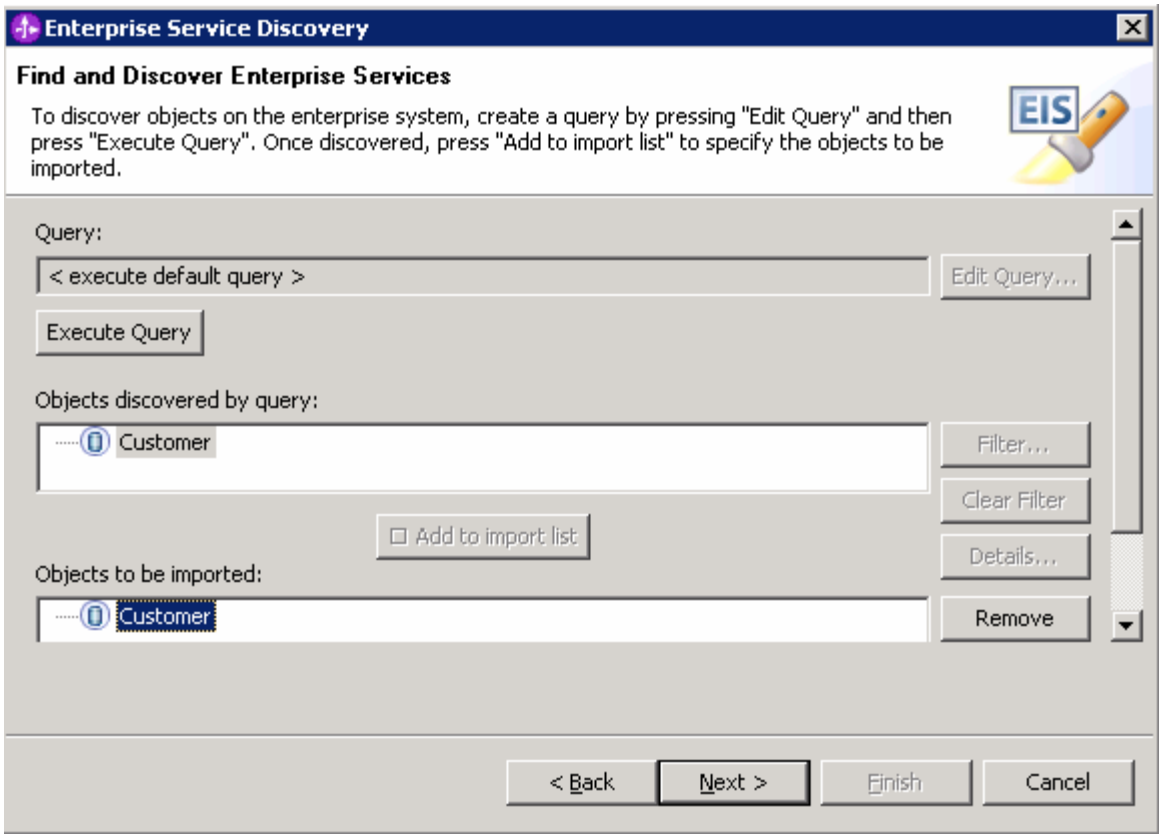
__ d. Click the **Show Advanced >>** button to see the Log file location and Logging level options for the discovery log and then click **Next** leaving the default log file location



____ 4. To Find and Discover the enterprise services, 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 a **Customer** business object under **Objects discovered by query**

__ b. Select **Customer** business object and click **Add to import list** button. The Customer business object will now be displayed under the **Objects to be imported** area

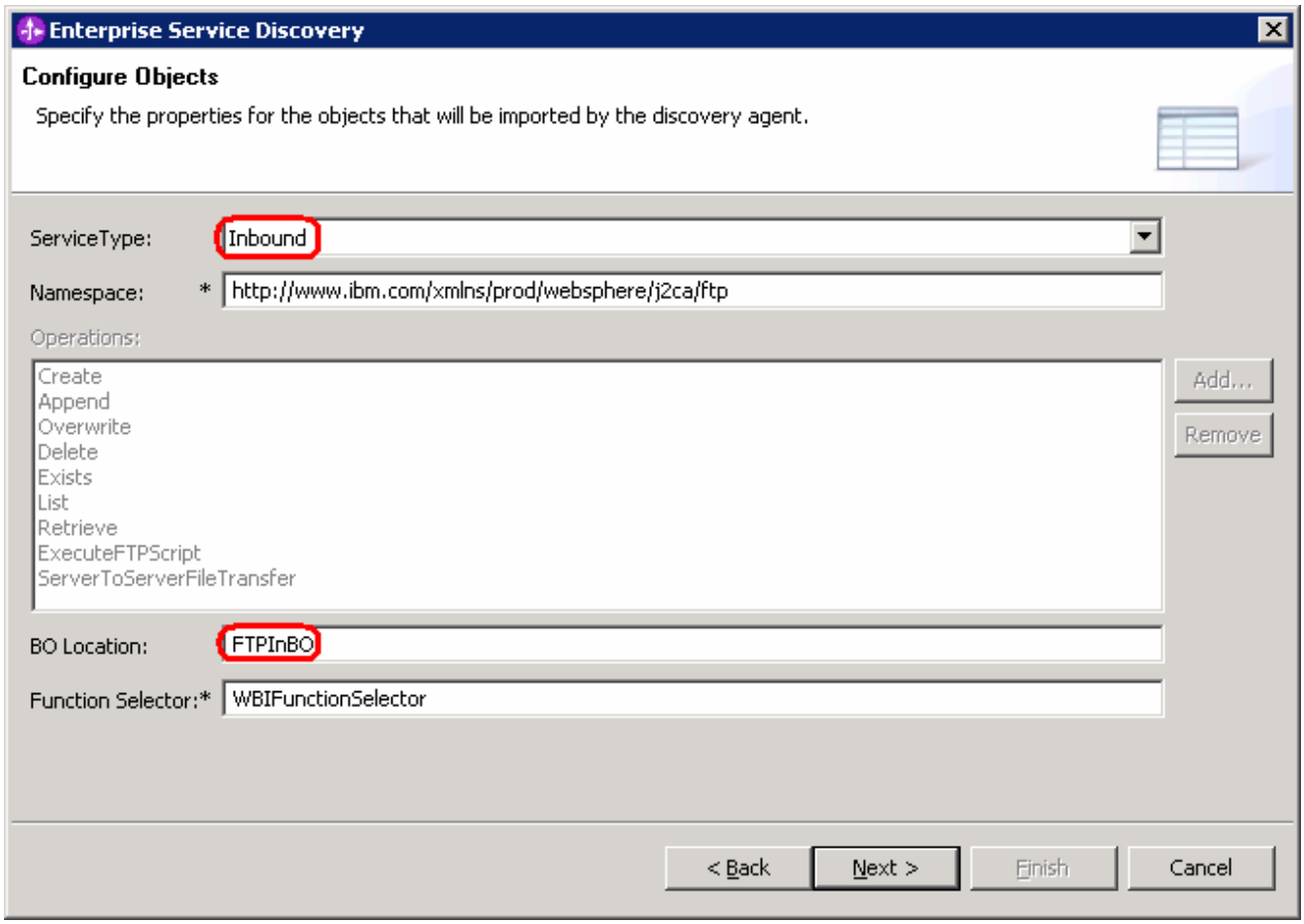


__ c. Click **Next**

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

__ a. From the Configure objects window, select **Inbound** from the dropdown list for the **Service Type** field

__ b. Enter **FTPInBO** for the **BO Location** field and click **Next**

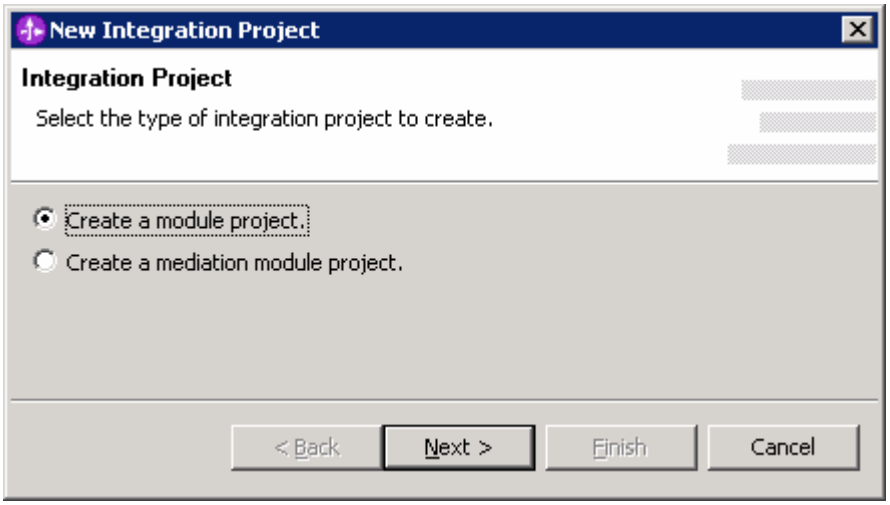


____ 6. Specify the properties for the artifacts that will be generated in your workspace

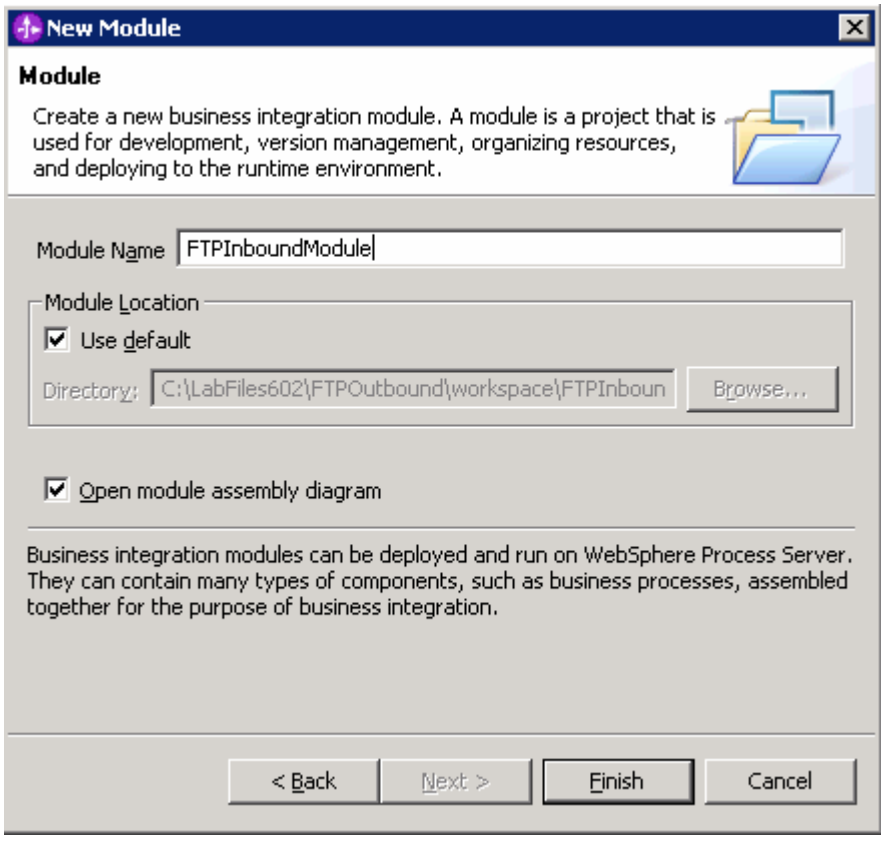
__ a. Create a new module

1) Click on the **New…** button next to the **Module** field

2) From the New Integration Project window, ensure that the radio button next to **Create a module project** is selected and click **Next**



3) Enter **FTPInboundModule** in the **Module Name** field and click **Finish**



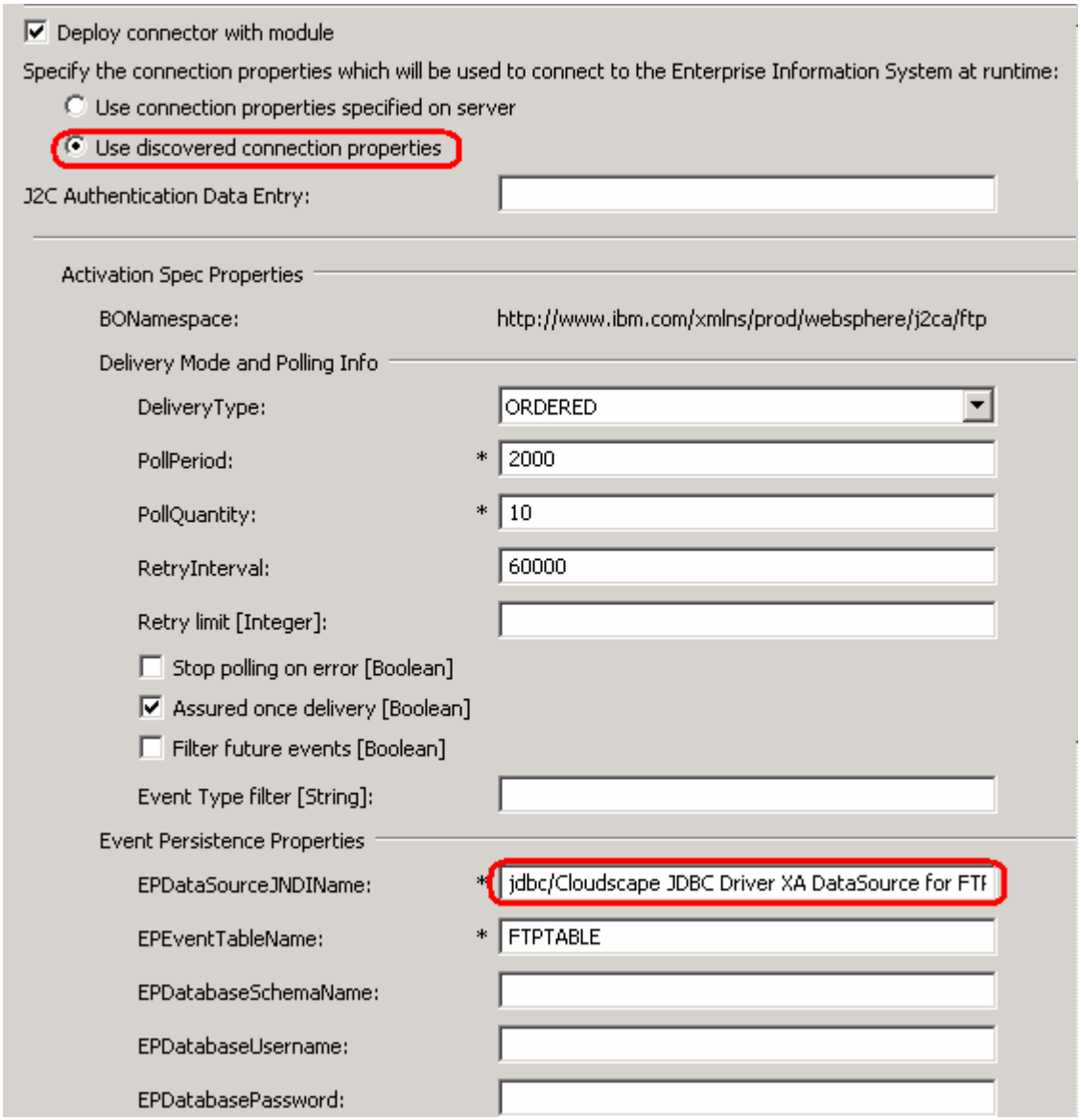
__ b. The module which is created above will appear under the **Module** field of the Generate Artifacts window

__ c. From the Generate Artifacts window, select the radio button next to **Use discovered connection properties**. This will make the Activation Spec Properties and Resource Adapter Properties visible

__ d. Enter the following for the **Event Persistence Properties** of the Activation Spec Properties:

1) EPDataSourceJNDIName: **jdbc/Cloudscape JDBC Driver XA DataSource for FTP**

2) Accept the default EPEventTableName: **FTPTABLE**



__ e. Enter the following for **FTP Adapter Properties** of Activation Spec Properties:

1) EventDirectory**: ftp://<FTP Server IP Address>/<EventDir>**

Note: EventDir is the event directory that you created on Step 2 of Part1

---

2) Username: **<user>** (username using which you connect to your FTP machine)

3) Password: **<password>** (password for the above user to connect to your FTP machine)

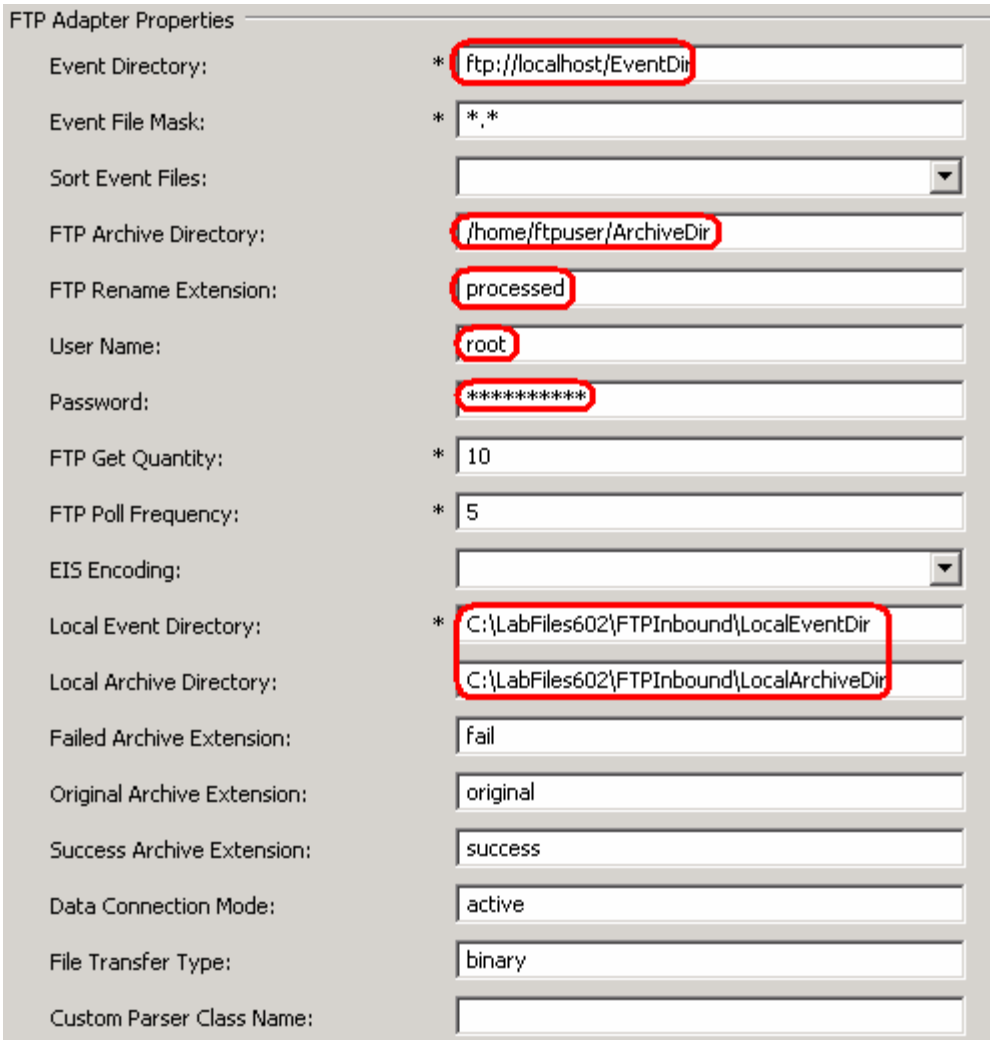
4) FTPArchiveDirectory: **<ArchiveDir>**

Note: ArchiveDir is the archive directory that you created on Step 2 of Part 1

5) FTPRenameExt: **processed**

6) LocalEventDirectory: **<LOCAL_EVENT_DIR>**

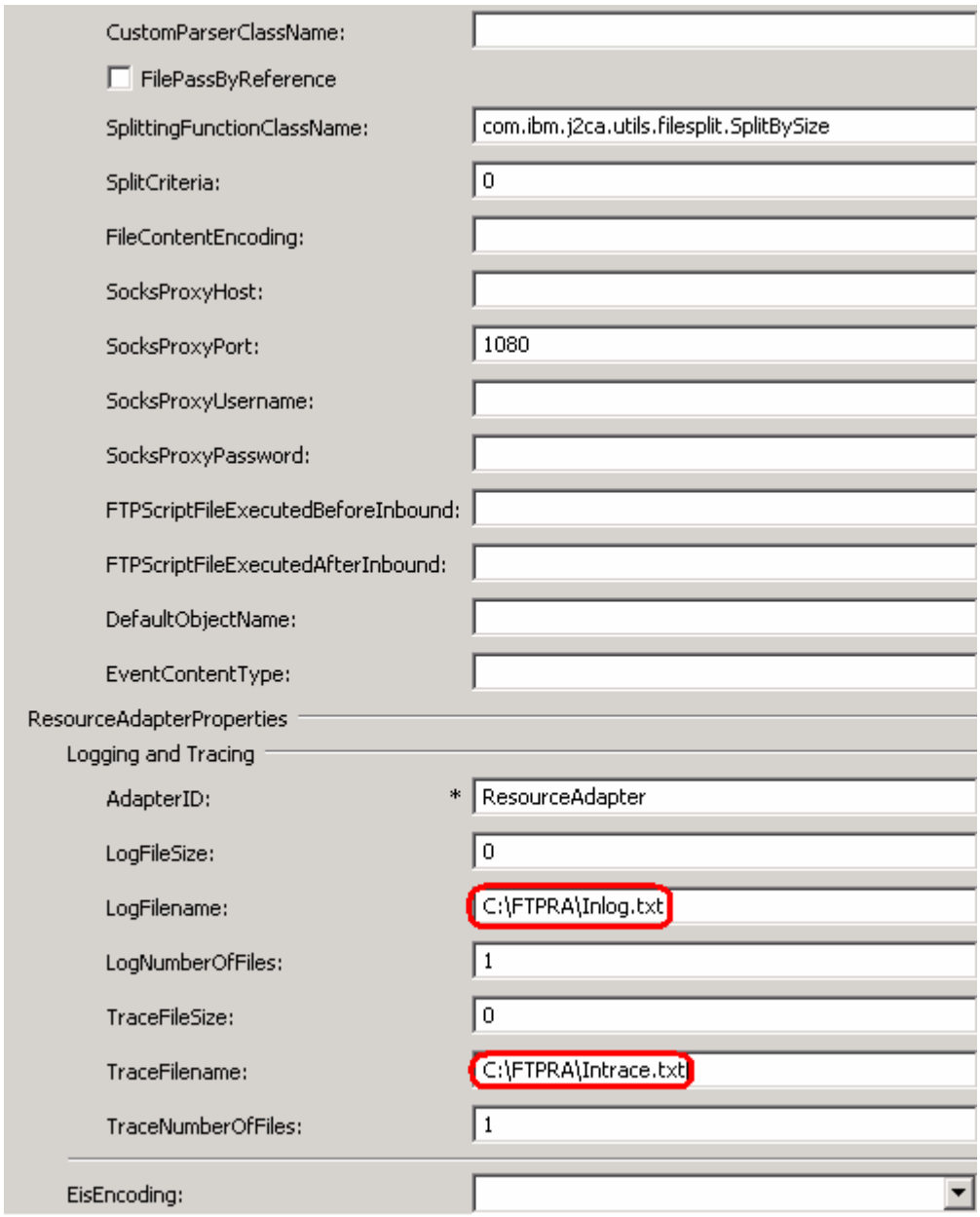
7) LocalArchiveDirectory: **<LOCAL_ARCHIVE_DIR>**



__ f. Enter the following for the **Logging and Tracing** properties of the **ResourceAdapterProperties**:

1) LogFilename: **C:\FTPRA\Inlog.txt**

2) TraceFilename: **C:\FTPRA\Intrace.txt**

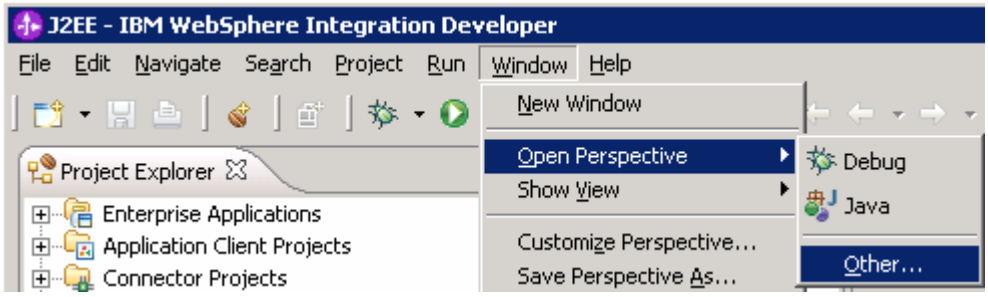


__ g. Click **Finish**

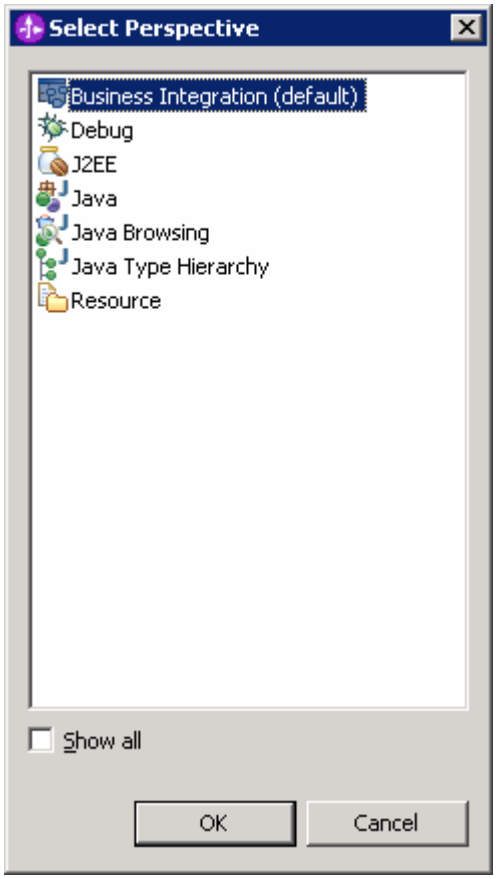
____ 7. Create a Java Component and wire it to **FTPInboundInterface**

__ a. Change to Business Integration perspective if not open

1) Select **Window > Open Perspective > Other…**.



2) From the Select Perspective window, select **Business Integration (default)** and click **OK**



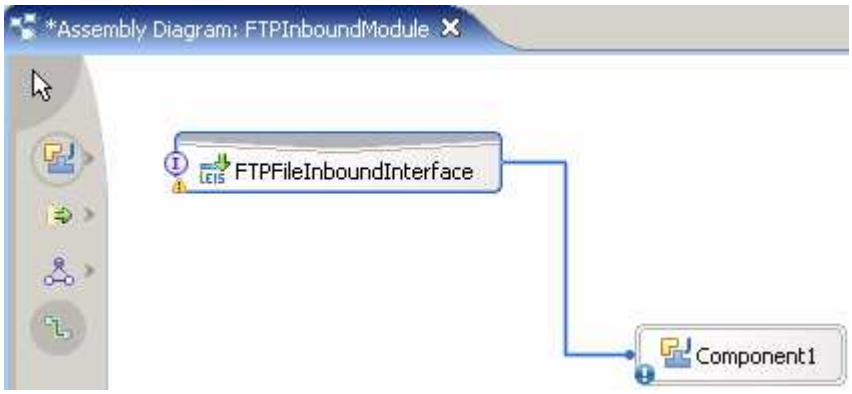
__ b. Expand **FTPInboundModule** and double-click **FTPInboundModule** to open it in the Assembly Editor

__ c. From the palette, click the **Component (with no implementation type)** icon, and select **Java** component from the context menu icon and then click in the Assembly Diagram's empty space to drop it there

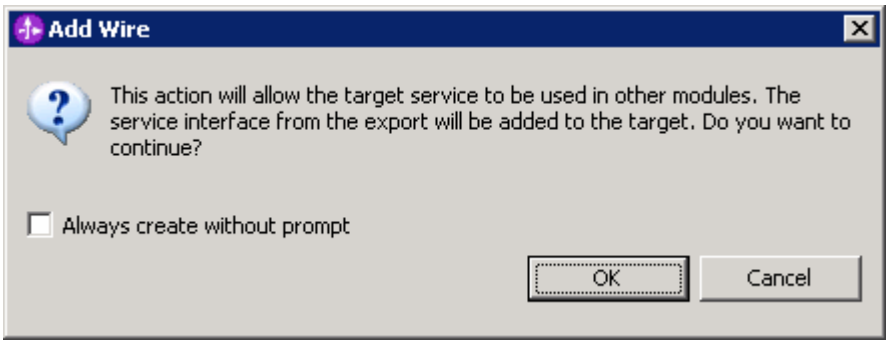
__ d. Wire the **FTPInboundInterface** to the **Component1**

1) Select the **wire** icon from the palette

2) Click on **FTPFileInboundInterface** and then click on **Component1** to wire them together



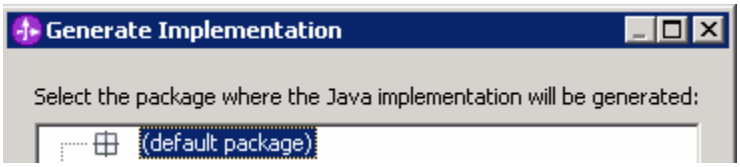
3) Select **OK** for the Add Wire popup window



4) Click on the **Selection Tool** to get back to the normal cursor mode

__ e. Right-click on **Component1** and select **Generate Implementation** from the context menu

__ f. Click **OK** from the Generate Implementation window to accept the **default package**



__ g. Component1Impl.java will be opened in Assembly editor. Scroll down to the method **emitFTPFile(DataObject emitFTPFileInput)** and **emitFTPFile(DataObject emitCustomerWrapperInput)** that needs to be implemented and add the code as shown below for each of those methods:

```
public void emitCustomerWrapper(DataObject emitCustomerWrapperInput) {
    //TODO Needs to be implemented.
    System.out.println("*****Reached End Point for NON PASS-THROUGH: CUSTOMER*****");
}

/**
 * Method generated to support implemention of operation "emitFTPFile" defined for WSDL port type
 * named "interface.FTPFileInboundInterface".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitFTPFile(DataObject emitFTPFileInput) {
    //TODO Needs to be implemented.
    System.out.println("*****Reached End Point for PASS-THROUGH: FTP*****");
}
```

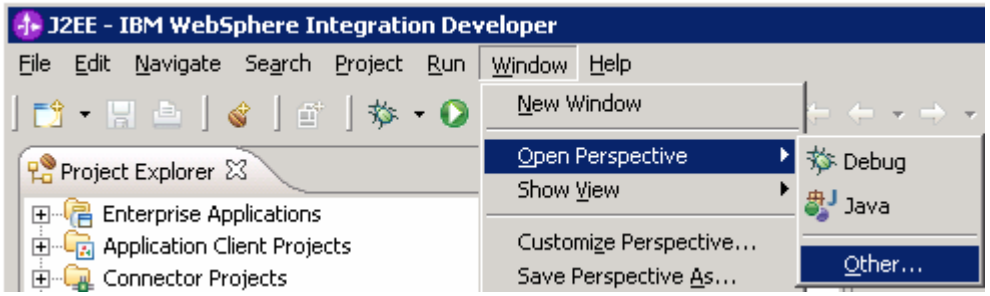
__ h. Save (**Ctrl + S**) and close Component1Impl.java editor

__ i. Save (**Ctrl + S**) and close Assembly diagram: FTPInboundModule

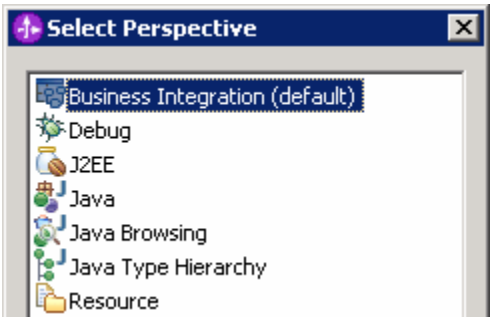
____ 8. You can also configure/change the adapter properties using the Assembly Editor

__ a. Change to Business Integration perspective if you are in a different perspective

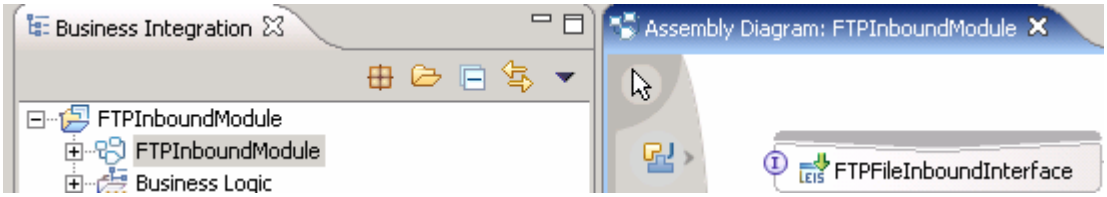
1) Select **Window > Open Perspective > Other…**.



2) From the Select Perspective window, select **Business Integration (default)** and click **OK**



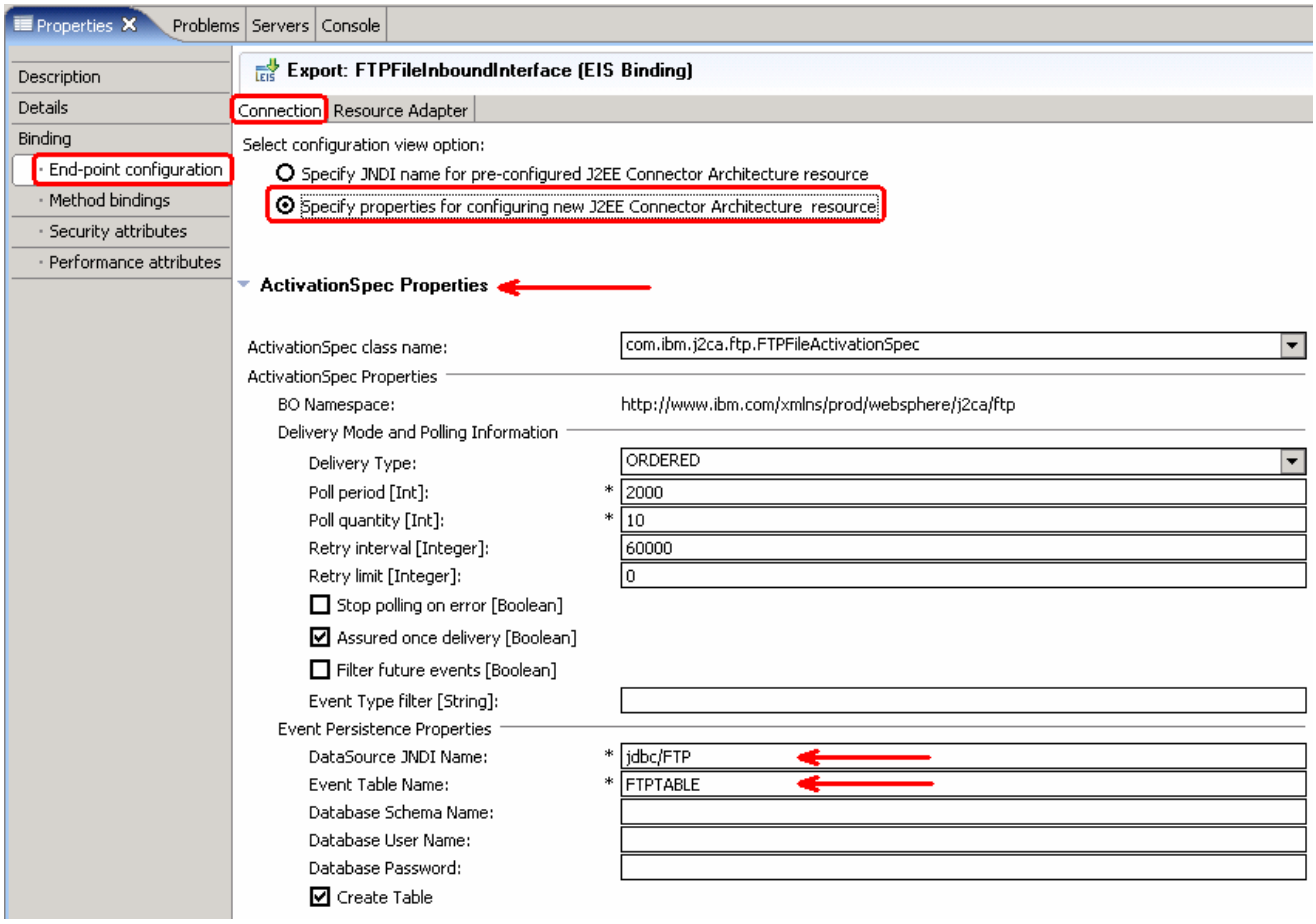
__ b. Expand **FTPInboundModule** and double-click **FTPInboundModule** to open it in Assembly Editor

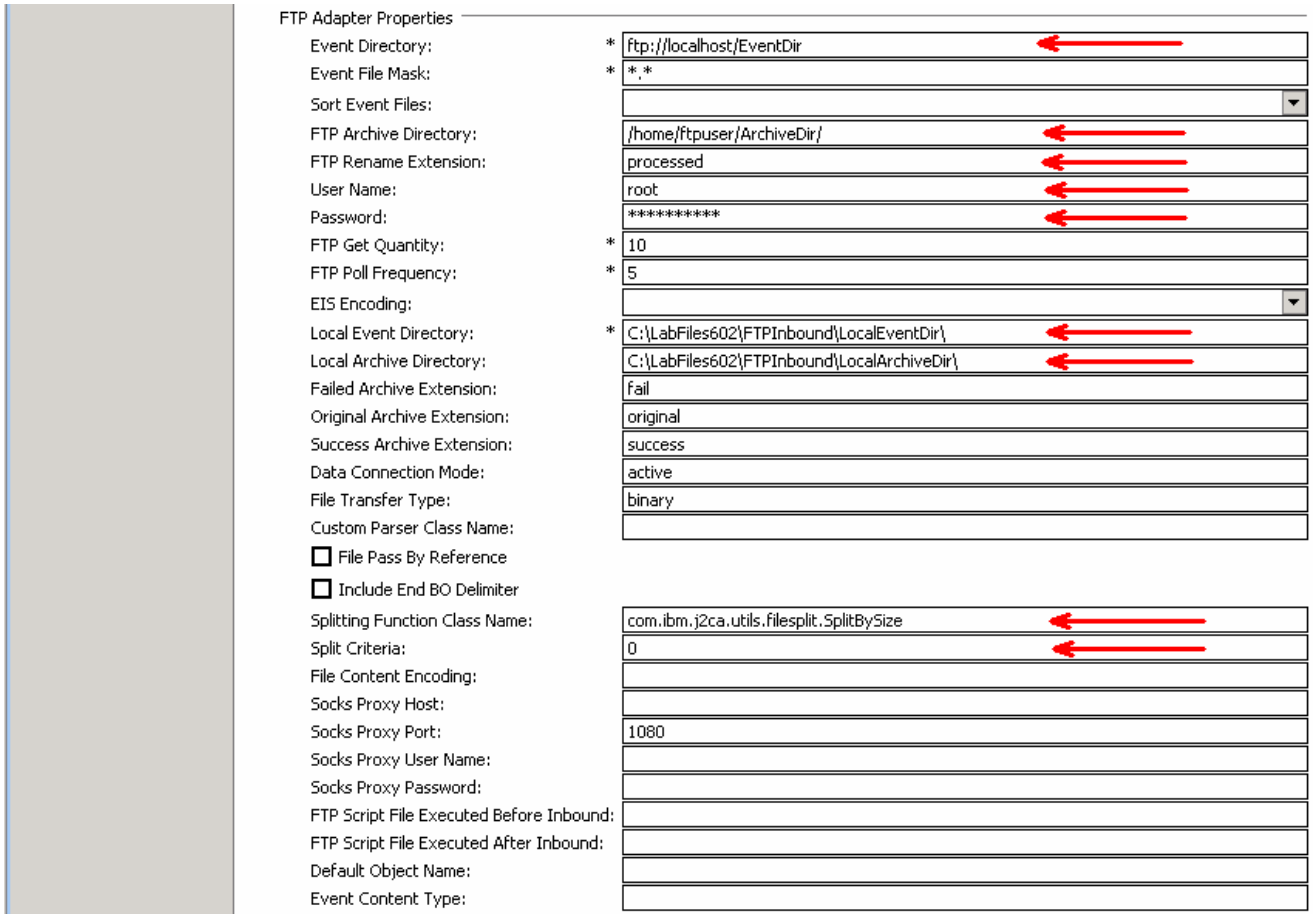


__ c. Click on **FTPFileInboundInterface** from the Assembly Editor and select **Properties** tab from the bottom

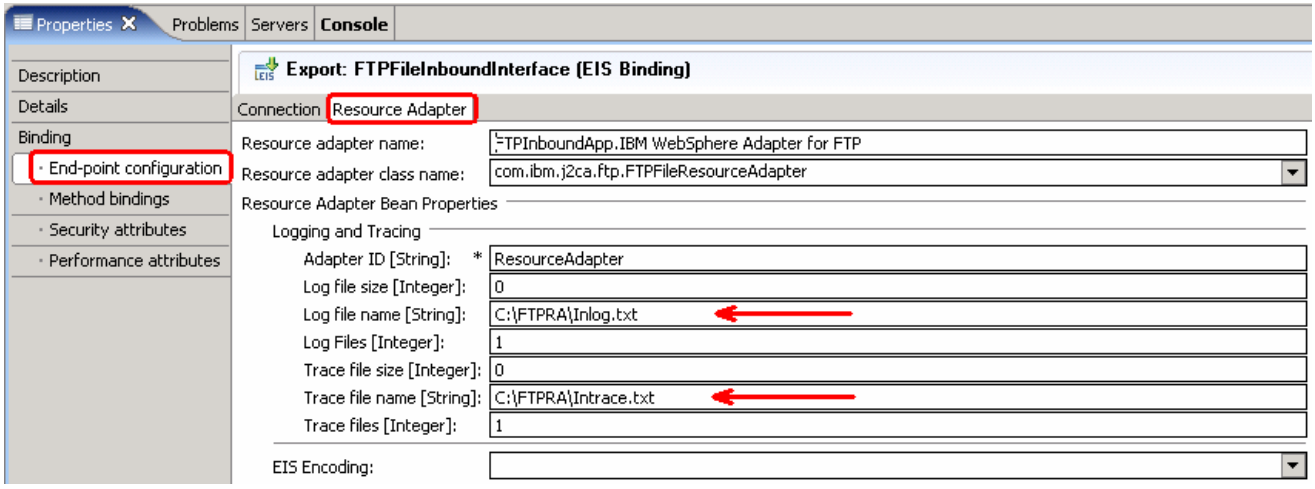
__ d. Select **Binding** under Properties and select **Endpoint Configuration** under Binding itself and then select the **Connection** tab

__ e. Ensure the radio button next to **Specify properties for pre-configured new J2EE Connector Architecture recourse** is selected and then **c**lick on **Activation Spec Properties** to expand them. You can change these properties, which you entered using the Enterprise Service Discovery (ESD) wizard in the previous steps, and make sure that you save those changes before you deploy the application onto the server





__ f. You can also select **Resource Adapter** tab and review/change those properties

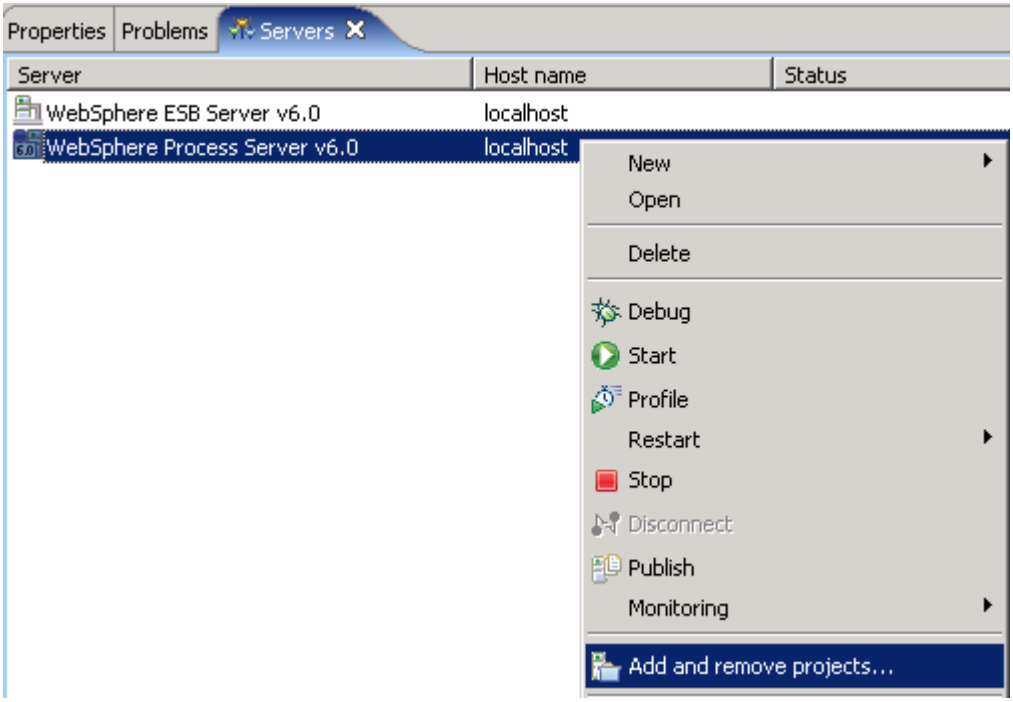


Part 5: Test Scenario: Simple PassThrough

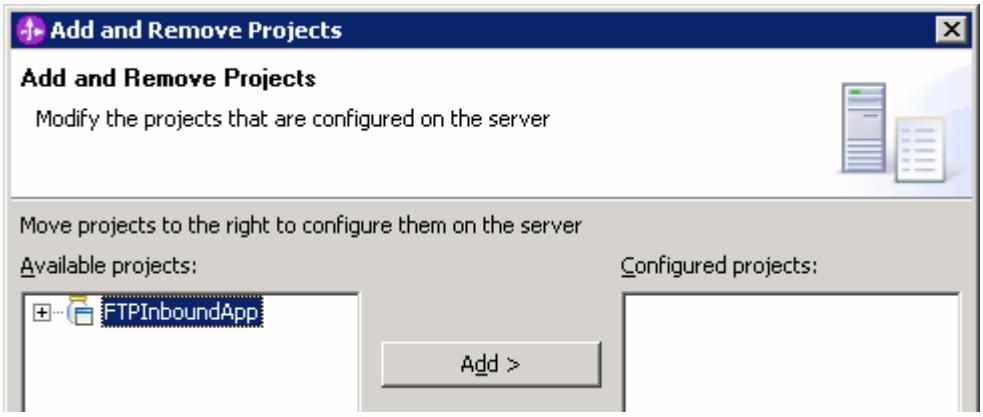
In this part of the lab, you will use the WebSphere Process Server Test Environment to test the SCA application Inbound processing.

____ 1. Add the project to the WebSphere Process Server Test Environment

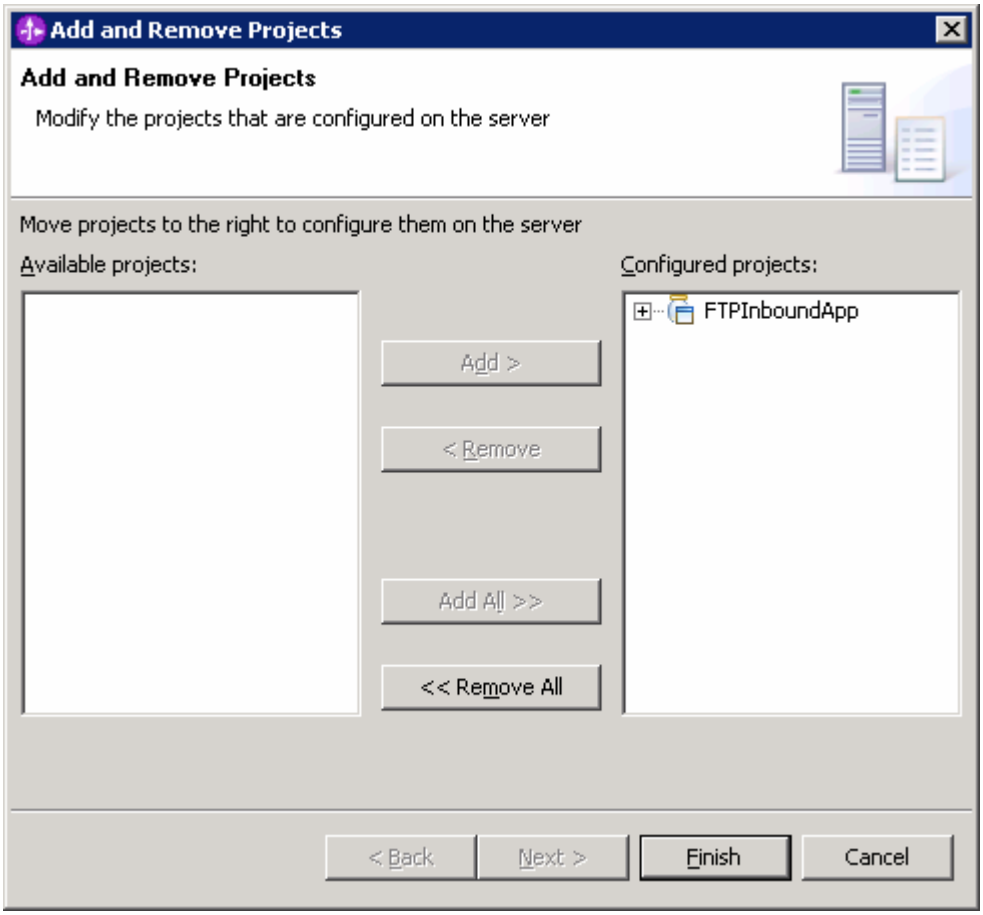
__ a. Right-click on **WebSphere Process Server v6.0** under the server view and **select Add and remove projects…** from the context menu



__ b. From the Add and Remove Projects window, select **FTPInboundModuleApp** under Available projects panel and click **Add >**



__ c. You will now see the **FTPInboundModuleApp** added to the **Configured projects**



__ d. Click **Finish**. 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

____ 2. Test the adapter application

__ a. On the machine where the FTP Server is running, put a test file in the **EventDir**. The file will be polled from the event directory on the FTP server and will be transferred to the local event directory. And from the local event directory, it will be moved to configured endpoint

---

For your convenience, a test file with the name **SplitBySize.txt** is placed under **<LAB_FILES>**

__ b. Check for the test file in **<LOCAL_EVENT_DIR>** folder on your local machine. The file quickly will be moved to local archive directory

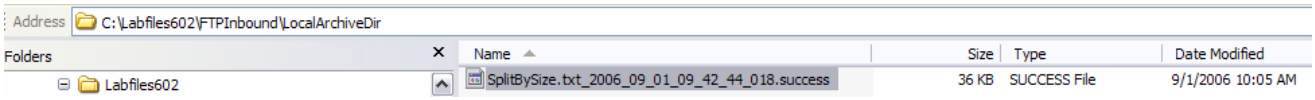
__ c. Check the **ArchiveDir** of your FTP server which should contain the same file name appended with year, month, date, system time, and processed as was given while configuring ESD

____ 3. Verify your results

__ a. Check your Test Environment console (or Systemout.log) for the following successful message:

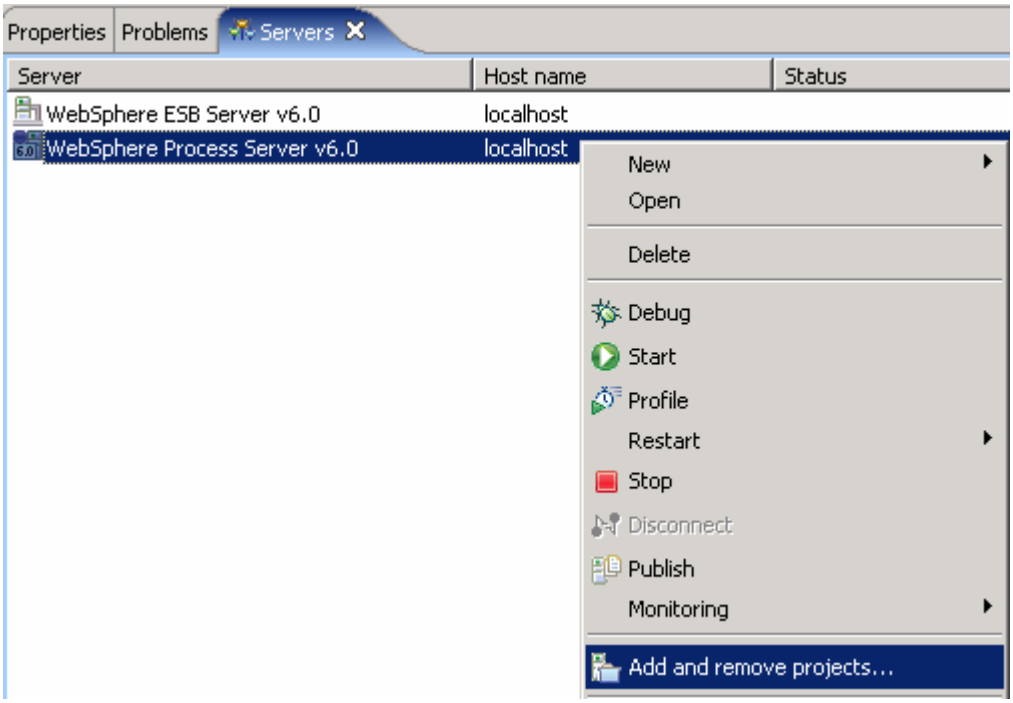
```
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for PASS-THROUGH: FTP*****
```

__ b. Check for the test file in **<LOCAL_ARCHIVE_DIR>** which should contain an archive of the event file, with the same file name appended with year, month, date, system time, and success

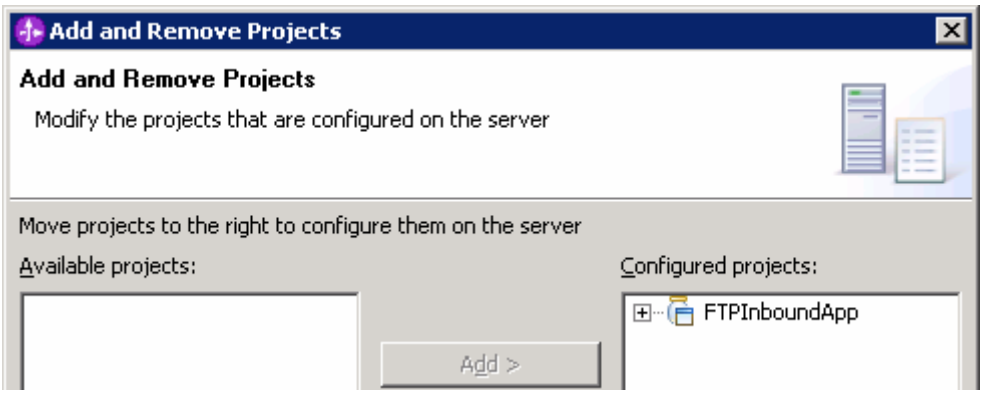


____ 4. Restore the Server Configuration

__ a. Right-click on **WebSphere Process Server v6.0** under the Servers view and select **Add and remove projects…** from the context menu

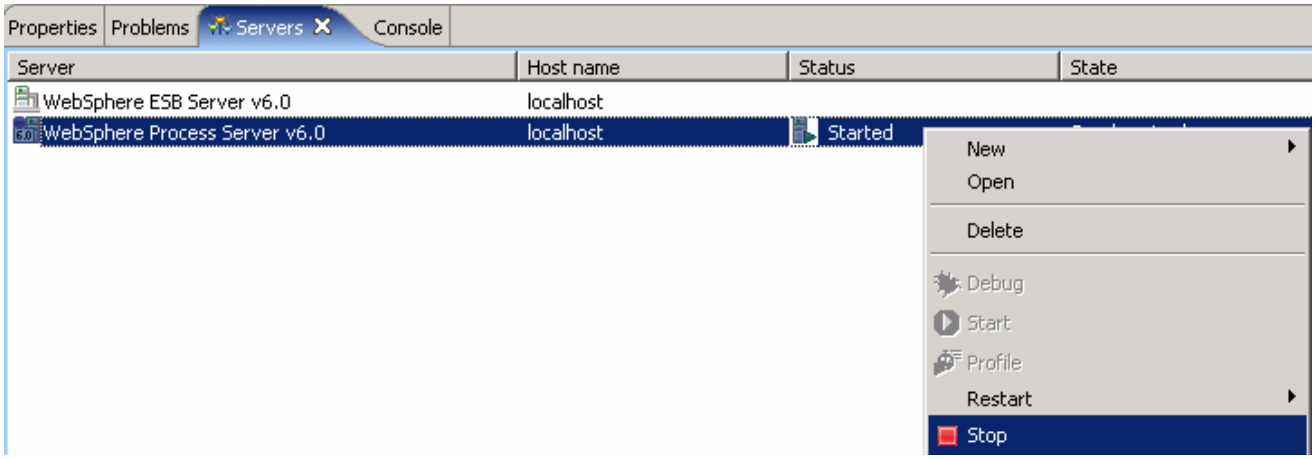


__ b. Select **FTPInboundModuleApp** under Configured projects and click **< Remove**



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

__ d. Right-click on **WebSphere Process Server v6.0** from the Servers view and select **Stop** from the context menu



Part 6: Test Scenario: Simple Data Transformation (Non Pass-Through)

In this part of the lab, you will edit the Activation Spec Properties to see how a simple Data Transformation works for an Inbound operation.

____ 1. Open the Activation Spec Properties from the Attributes View

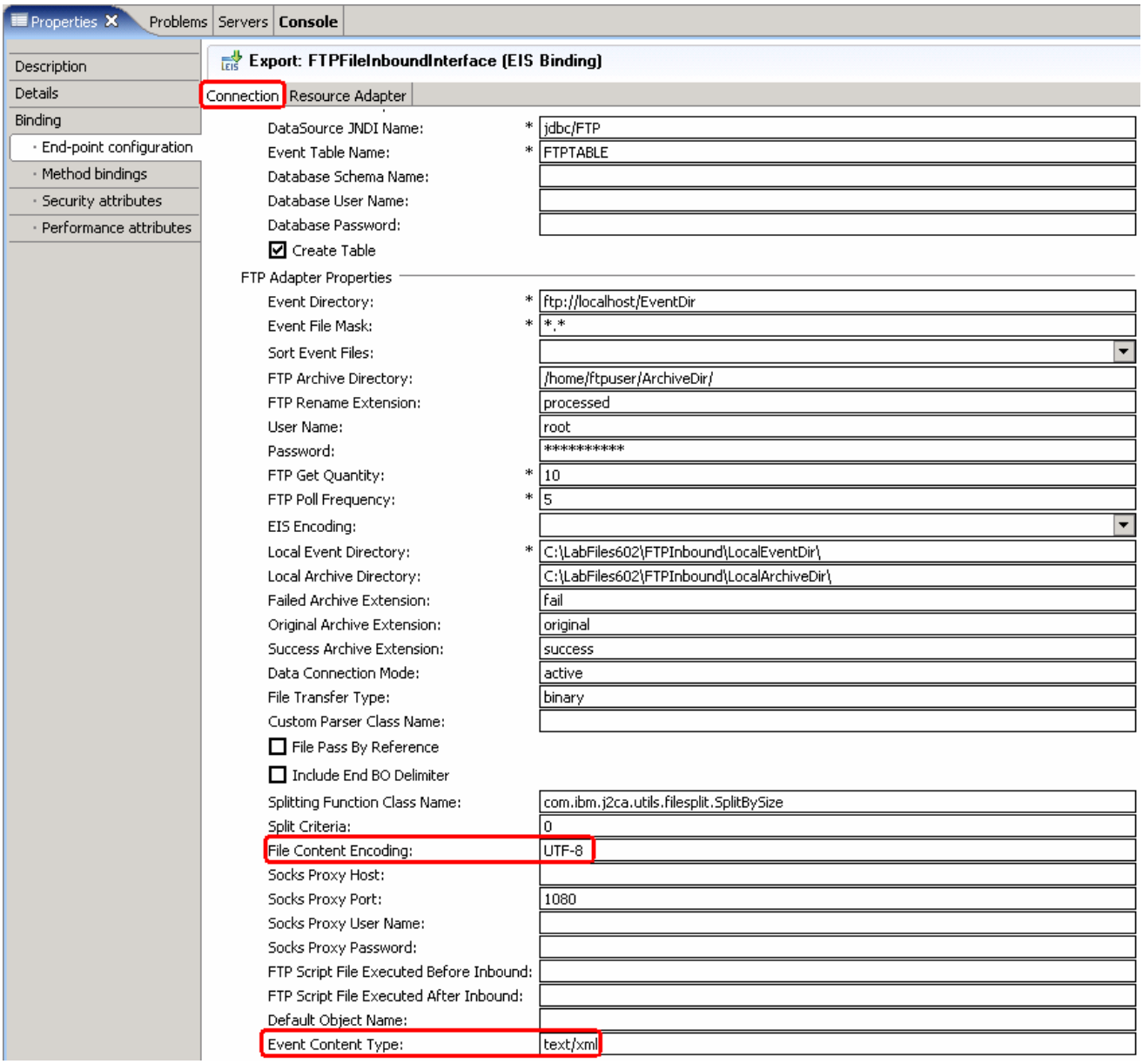
__ a. Click on **FTPFileInboundInterface** from the Assembly Editor and select **Properties** tab from the bottom

__ b. Select **Binding** under Properties and select **Endpoint Configuration** under Binding itself and then select the **Connection** tab

__ c. Ensure the radio button next to **Specify properties for pre-configured new J2EE Connector Architecture recourse** is selected and then **c**lick on **Activation Spec Properties** to expand them and enter the following fields:

1) FileContentEncoding: **UTF-8** (or any other valid encoding)

2) EventContentType: **text/xml**



__ d. Save (**Ctrl +S**) your changes

____ 2. Follow step 1 of Part 5 to add the modified **FTPInboundModuleApp** to the server

____ 3. Test the adapter

__ a. On the machine where the FTP Server is running, put a **SimpleDTF.xml** file in the **EventDir**

Note: For your convenience, a **SimpleDTF.xml** file is placed in **<LAB_FILES>\FTPFiles**. The file has one Business Object.

__ b. Check for the **SimpleDTF.xml** file in **<LOCAL_EVENT_DIR>** folder on your local machine. The xml file will be quickly moved to the local archive directory

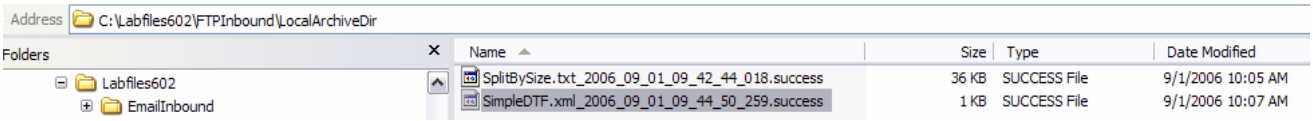
__ c. Check the **ArchiveDir** of your FTP server which should contain the same file name appended with year, month, date, system time, and processed as was given while configuring ESD

____ 4. Verify your results

__ a. Check your Test Environment console (or SystemOut.log file) for the following successful message:

```
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for NON PASS-THROUGH: CUSTOMER*****
```

__ b. Check your **<LOCAL_ARCHIVE_DIR>** which should contain an archive of the event file, with the same file name appended with year, month, date, system time, and success



____ 5. Repeat step 4 of **Part 5**, to restore server configuration

Part 7: Test Scenario: SplitByDelimiter for Data Transformation

In this part of the lab, you will edit the Activation Spec Properties to see how **SplitByDelimiter** works with Data Transformation for an Inbound operation.

____ 1. Open the Activation Spec Properties from the Attributes View

__ a. Click on **FTPFileInboundInterface** from the Assembly Editor and select **Properties** tab from the bottom

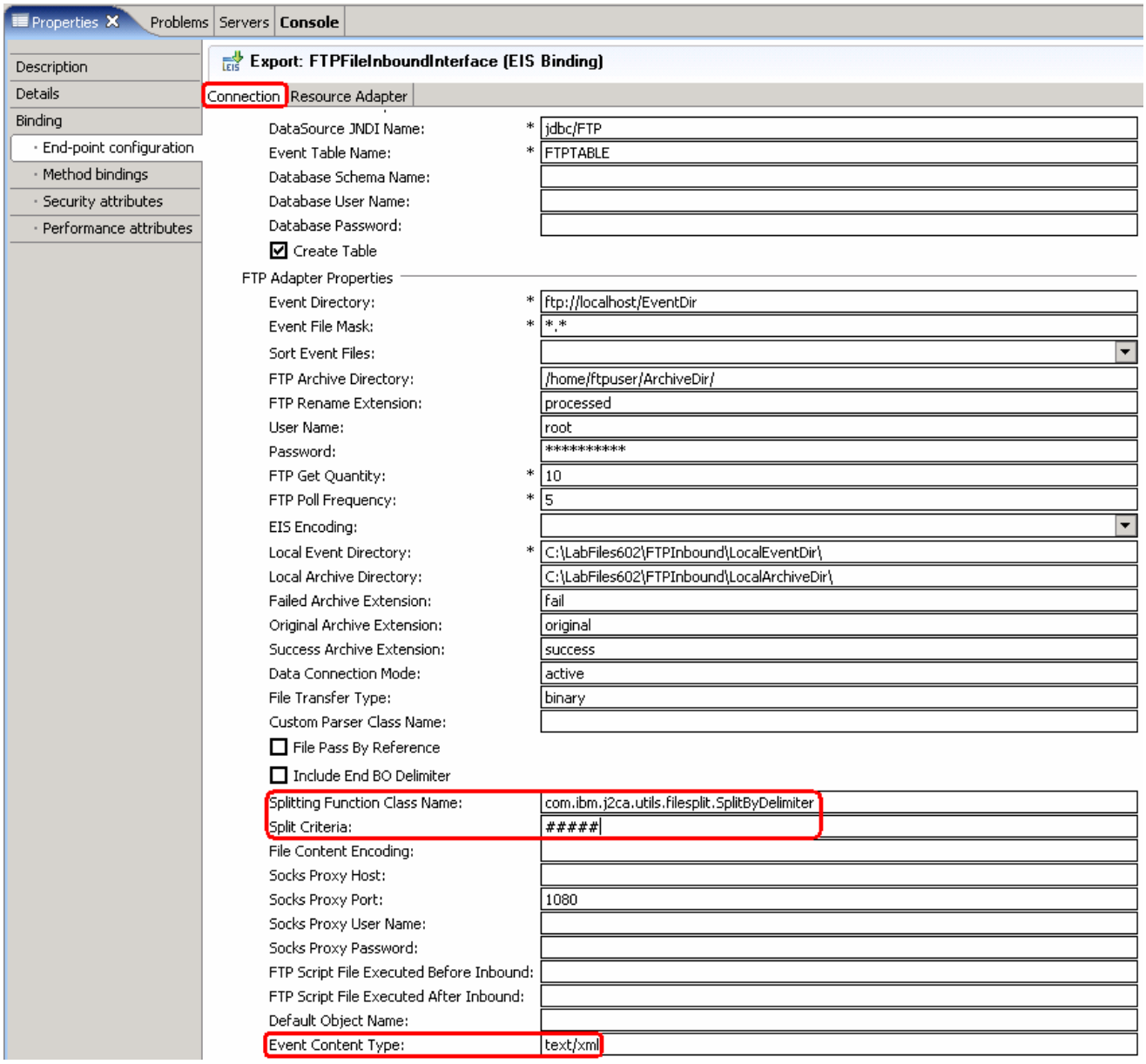
__ b. Select **Binding** under Properties and select **Endpoint Configuration** under Binding itself and then select the **Connection** tab

__ c. Ensure the radio button next to **Specify properties for pre-configured new J2EE Connector Architecture recourse** is selected and then **c**lick on **Activation Spec Properties** to expand them. You can change these properties that you entered using the Enterprise Service Discovery wizard in the previous steps and save those changes before you deploy the application onto the server

1) SplittingFunctionClassName: **com.ibm.j2ca.utils.filesplit.SplitByDelimiter**

2) SplitCriteria: #####

3) EventContentType: **text/xml**



__ d. Save (**Ctrl + S**) your changes

____ 2. Add the modified **FTPInboundModuleApp** as explained in Step 1 of **Part 5**

____ 3. Test the application

__ a. On the machine where the FTP Server is running, put a **SplitByDelimiter.xml** file in the **EventDir**

Note: For your convenience, a **SplitByDelimiter.xml** file is placed in **<LAB_FILES>\FTPFiles**. The file has two Business Objects separated by the delimiter #####.

__ b. Check for the **SplitByDelimiter.xml** file in **<LOCAL_EVENT_DIR>** folder on your local machine. The xml file will be quickly moved to the local archive directory

__ c. Check the **ArchiveDir** of your FTP server which should contain the same file name appended with year, month, date, system time, and processed as was given while configuring ESD

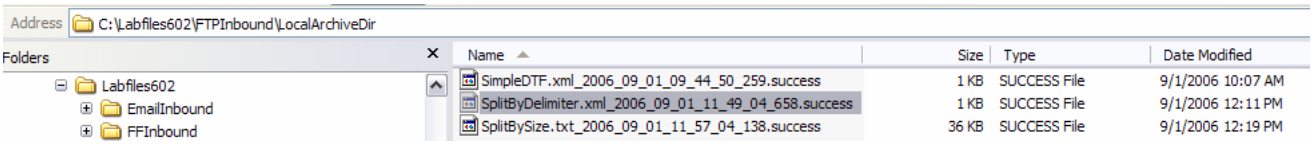
____ 4. Verify your results

__ a. Check your Test Environment console (or Systemout.log file) for the following successful message:

```
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for NON PASS-THROUGH: CUSTOMER*****
SystemOut     O *****Reached End Point for NON PASS-THROUGH: CUSTOMER*****
```

Note: You will see the successful event delivery message twice as there were two Business Objects present in the event file separated by the delimiter #####.

__ b. Check your **<LOCAL_ARCHIVE_DIR>** which should contain an archive of the event file, with the same file name appended with year, month, date, system time, and success



____ 5. Repeat step 4 of **Part 5**, to restore server configuration

Part 8: Test Scenario: SplitBySize for Pass Through

In this part of the lab, you will edit the Activation Spec Properties to see how SplitBySize works for an Inbound operation.

____ 6. Open the Activation Spec Properties from the Attributes View

__ a. Click on **FTPFileInboundInterface** from the Assembly Editor and select **Properties** tab from the bottom.

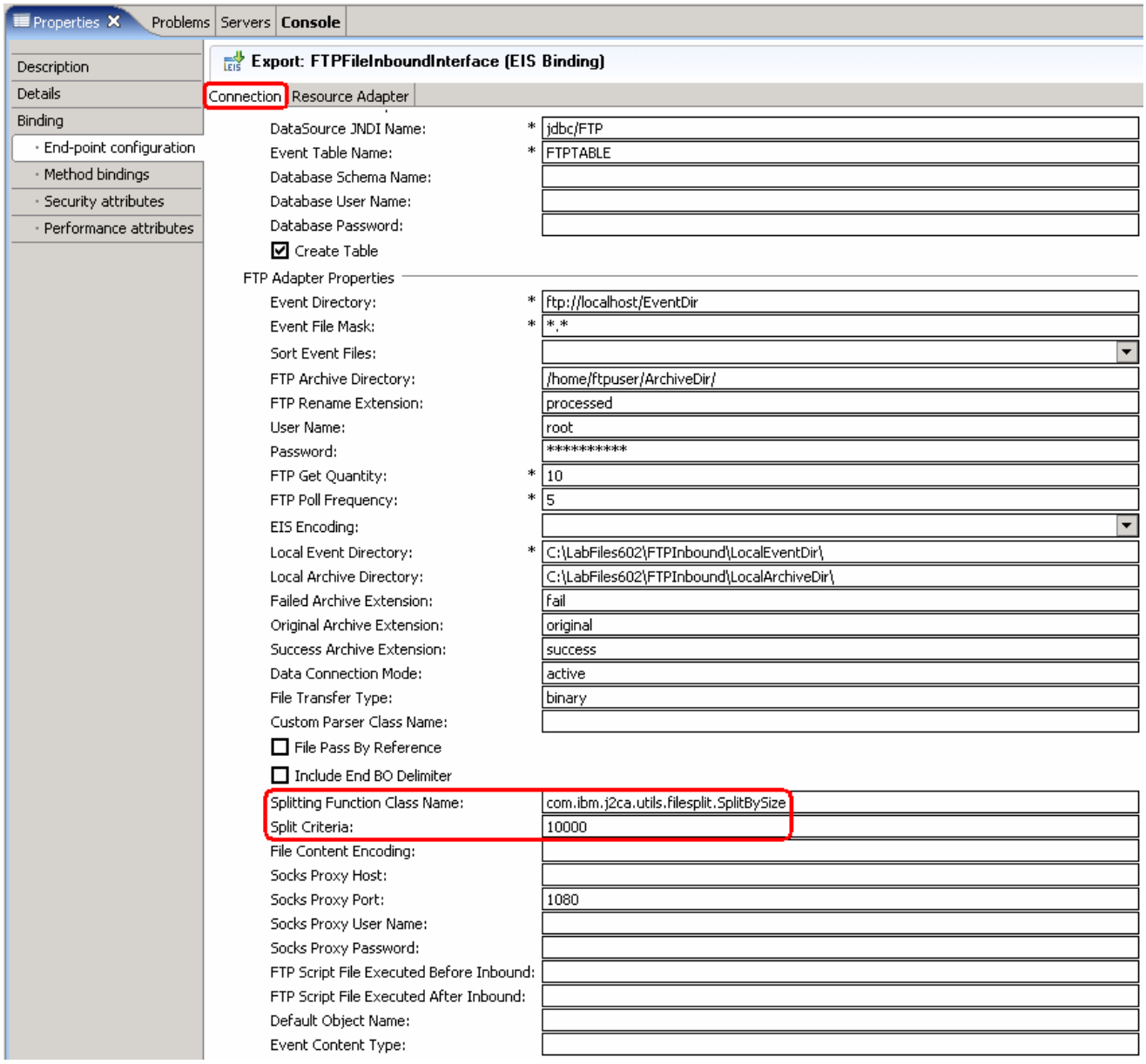
__ b. Select **Binding** under Properties and select **Endpoint Configuration** under Binding itself and then select the **Connection** tab

__ c. Ensure the radio button next to **Specify properties for pre-configured new J2EE Connector Architecture recourse** is selected and then **c**lick on **Activation Spec Properties** to expand it and enter the following fields:

1) SplittingFunctionClassName: **com.ibm.j2ca.utils.filesplit.SplitBySize**

2) SplitCriteria: **10000**

3) Make sure that you delete any values you have for **FileContentEncoding** and **EventContentType** fields.



__ d. Save (**Ctrl +S**) your changes

____ 7. Follow step 1 of **Part 5** to add the modified FTPInboundModuleApp to the server

____ 8. Test the adapter

Note: For your convenience, a test file **SplitBySize.txt** is placed in **<LAB_FILES>**. Note the size of the file.

__ a. On the machine where the FTP Server is running, put a **SplitBySize.txt** file in the **EventDir**

__ b. Check for the **SplitBySize.txt** file in **<LOCAL_EVENT_DIR>** folder on your local machine. The text file will be quickly moved to the local archive directory

__ c. Check the **ArchiveDir** of your FTP server which should contain the same file name appended with year, month, date, system time, and processed as was given while configuring ESD

____ 9. Verify your results

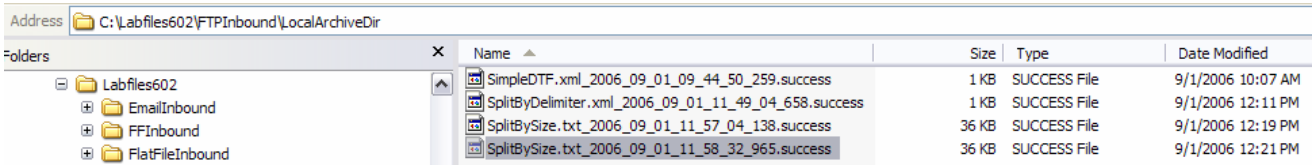
__ a. Check your test environment console (or Systemout.log file) for the following successful message:

Note: You will see the successful event delivery message for four times as the file size is 37KB and the split criteria has 10KB, which means it splits the file into four part and delivers to the endpoint.

```
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for PASS-THROUGH: FTP*****
SystemOut     O *****Reached End Point for PASS-THROUGH: FTP*****
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for PASS-THROUGH: FTP*****
ResourceAdapt A com.ibm.j2ca.extension.eventmanagement.internal.EventSender deliverEvent() CWYBS0505I: The event has been delivered.
SystemOut     O *****Reached End Point for PASS-THROUGH: FTP*****
```

Note: The same SplitBySize.txt file was delivered as one event file to the end point when you did not specify the SplitCriteria in part 5 while configuring ESD. Here, in this part, you will see the successful event delivery message for four times for the same file (with the file size being 36KB) because of the split criteria that has been given as 10KB, which means it splits the file into four part and delivers to the endpoint.

__ b. Check your **<LOCAL_ARCHIVE_DIR>** which should contain an archive of the event file, with the same file name appended with year, month, date, system time, and success



____ 10. Repeat step 4 of **Part 5**, to restore server configuration.

What you did in this exercise

In this lab, you created the database in Cloudscape required for this lab. You continued with importing the FTP Adapter RAR file into your WebSphere Integration Developer workspace. Then, used your WebSphere Process Server administrative console and configured it to create Data source required to complete this lab.

You made use of Enterprise Service Wizard available in WebSphere Integration Developer to specify Activation Spec Properties and Resource Adapter Properties which, after deploying onto the server will generate Business Objects and other artifacts.

In the end you deployed and then tested the adapter application for two pass-through test scenarios – simple pass through and split by size and two data transformation test scenarios – simple data transformation and split by delimiter.