

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

## E-mail Adapter inbound lab

What this exercise is about .....	1
Lab requirements .....	1
What you should be able to do .....	2
Introduction .....	3
Exercise instructions .....	5
Part 1: Initialize the workspace and prepare for the lab.....	7
Part 2: Pass through scenario.....	8
2.1.  Configure pass through using the external service wizard .....	9
2.2.  Add Java™ component.....	18
2.3.  Test pass through scenario.....	20
Part 3: Content specific (non-pass through) scenario.....	24
3.1.  Configure non-pass through using the external service wizard.....	25
3.2.  Add Java component .....	36
3.3.  Test non pass through scenario.....	39
Part 4: User defined (Fixed structure e-mail) scenario .....	43
4.1.  Configure fixed structure using the external service wizard .....	44
4.2.  Add Java component .....	58
4.3.  Test user defined scenario.....	60
What you did in this exercise .....	63
Appendix: Define function selector .....	64
Task: Adding remote server to WebSphere Integration Developer test environment .....	66

### What this exercise is about

The objective of this lab is to provide you with an understanding of WebSphere Adapter for E-mail and inbound event processing. In this lab you will deploy the WebSphere Adapter for E-mail, using WebSphere Integration Developer, and integrate it with an SCA application that polls for inbound events and processes those inbound requests from the file system.

### Lab requirements

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

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- WebSphere Integration Developer V6.1 installed and updated with latest fixes
- WebSphere Process Server V6.1 Test Environment installed and updated latest fixes
- Install and configure the hMailServer (your e-mail server for this lab) and Mozilla Thunderbird (your e-mail client for this lab) by following the instructions in the lab:  
**WBIV61\_IEA\_AdapterConfigureEMailServerAndClientLab**
- Extract LabFiles61.zip to your C:\ (your root) drive

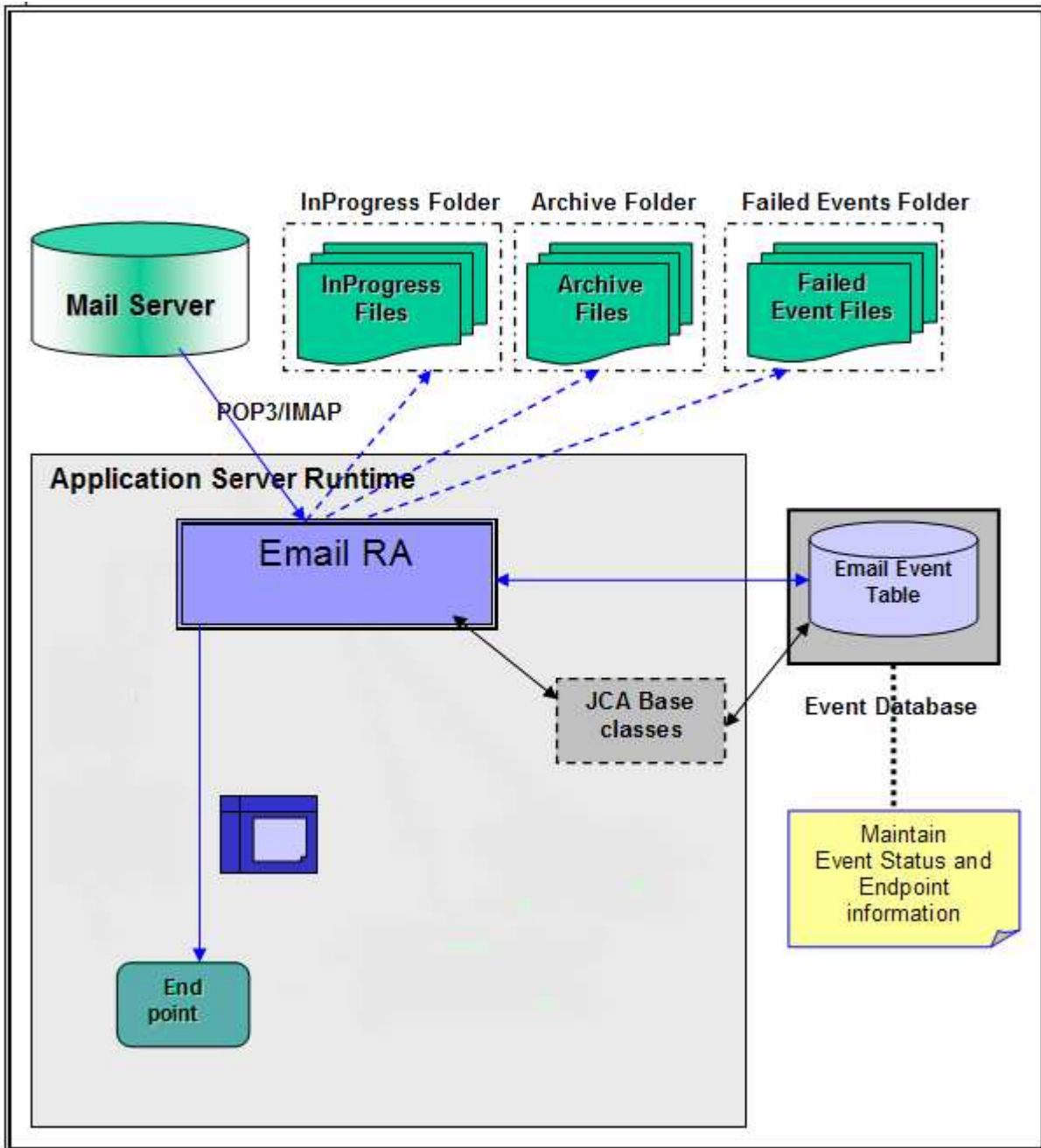
### What you should be able to do

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

- Import E-mail adapter RAR file into WebSphere Integration Developer
  - Use External Service wizard to configure Activation Spec Properties, Resource Adapter Properties to generate Business Objects and other artifacts and configure Function Selector, Data Binding and Data Handlers
  - Deploy the adapter application onto WebSphere Process Server
  - Test the deployed application using WebSphere Process Server test environment for pass-through, non pass-through, and Fixed structure e-mail scenarios
  - Restore the server configuration
-

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

Introduction



The E-mail RA interacts with the mail server using JavaMail API that handles the underlying SMTP and IMAP/POP3 protocols. The adapter supplies the necessary parameters required by JavaMail to communicate to the mail server.

The E-mail RA polls on the PollFolders for that user on the specified mail server at regular intervals, and picks up all the unread mails as events. You can specify multiple e-mail folders for a user's mail account within the PollFolders property. The multiple e-mail folders are then polled sequentially in the same poll cycle. Optionally, you can specify search criteria to pick up event mails and when search criteria is specified, all unread mails meeting the search criteria, are picked up for polling. When no search criteria are specified, all unread mails are picked up by default.

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

The E-mail RA writes all the polled mails to the InProgress file -folder, as files saved in the RFC822 format. The name of the RFC822 format file is the Message\_ID of the polled E-mail. The E-mail RA transforms the mail to an E-mail BO and delivers the same to configured endpoint. The event management framework takes care of delivering the event only once to the endpoint.

The contents of each E-mail are parsed into an EmailRecord object. The EmailRecord is routed through the Function Selector to the E-mail-specific data-binding. The E-mail-specific data-binding is aware of the structure of the E-mail BO. It will parse the EmailRecord structure, and route the content of the mail and the attachments to the mime-specific data-bindings, according to their mime-types. It then receives back the parsed structures and combine them back into the EmailBO, and send it out to the configured endpoint.

The E-mail Event Database is part of the Event Persistence Framework. The event persistence feature ensures that the event is delivered to the end point once and only once.

Once the mail is read and stored in the InProgress folder, it is deleted from the mail server (per the POP3 specifications). Once a mail is processed, the mail is explicitly marked as deleted on the Mail Server (per the IMAP specifications). The mail is archived or deleted from the InProgress folder. The archiving is based on the values for the properties ArchiveFolder and the FailedEventsFolder.

If the ArchiveFolder is specified, the successfully processed mails are moved into the ArchiveFolder from the InProgressFolder. If blank, then the successfully processed mails are just deleted from the InProgressFolder.

If the FailedEventsFolder is specified, the mails that failed processing are moved into the FailedEventsFolder from the InProgressFolder. If blank, then the failed mails are just deleted from the InProgressFolder.

---

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

**Exercise instructions**

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

Reference variable	Windows location	AIX® or UNIX® location
<WID_HOME>	C:\Program Files\IBM\WID61	
<WPS_HOME>	C:\<WID_HOME>\runtimes\bi_v6	
<EMAILADAPTER_HOME>	<WID_HOME>\ResourceAdapters\Email_6.1.0.0\deploy	
<LAB_FILES>	C:\Labfiles61	/tmp/Labfiles61
<WORKSPACE>	<LAB_FILES>\EmailInbound\workspace	
<EMAILFILES>	<LAB_FILES>\EmailFiles	
<STAGING>	<LAB_FILES>\EmailInbound\StagingDir	
<ARCHIVE>	<LAB_FILES>\EmailInbound\ArchiveDir	
<FAILED>	<LAB_FILES>\EmailInbound\FailedEventDir	
<TEMP>	C:\temp	/tmp

---

**Windows users note:** When directory locations are passed as parameters to a Java program such as EJBdeploy or wsadmin, it is necessary to replace the backslashes with forward slashes to follow the Java convention. For example, replace C:\LabFiles61\ with C:/LabFiles61/

---

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

**Instructions if using a remote server for testing**

Note that the previous table is relative to where you are running WebSphere Integration Developer. The following table is related to where you are running the remote test environment:

Reference variable	Example: Remote Windows test server location	Example: Remote z/OS <sup>®</sup> test server location	Input your values for the remote location of the test server
<SERVER_NAME>	server1	sssr011	
<WAS_HOME>	C:\Program Files\IBM\WebSphere\AppServer	/etc/sscell/AppServer	
<HOSTNAME>	localhost	mvsxxx.rtp.raleigh.ibm.com	
<SOAP_PORT>	8880	8880	
<TELNET_PORT>	N/A	1023	
<PROFILE_NAME>	AppSrv01	default	
<USERID>	N/A	ssadmin	
<PASSWORD>	N/A	fr1day	

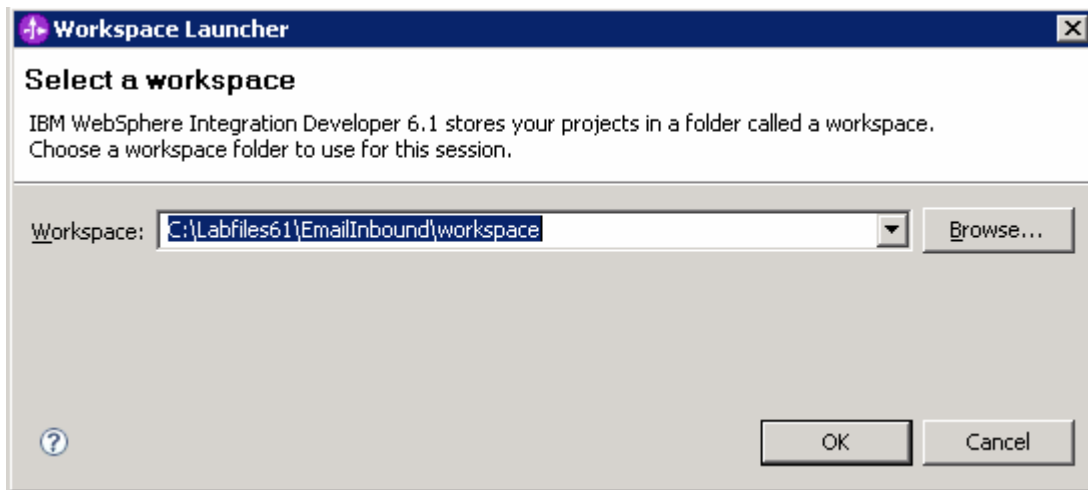
Instructions for using a remote testing environment, such as z/OS, AIX or Solaris, can be found at the end of this document, in the section "[Task: Adding remote server to WebSphere Integration Developer test environment](#)".


## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

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

This part of the lab, you will start the WebSphere Integration Developer V6.1 with a new workspace and create required data source and database using the administrative console of WebSphere Process Server V6.1

- \_\_\_ 1. Extract the provided LabFiles61.zip to your C:\ (root) drive, if you have not already done so. This will create the necessary subdirectory structure to complete the lab, and provides you with sample text files.
- \_\_\_ 2. Start the WebSphere Integration Developer V6.1 with a new workspace
  - \_\_\_ a. Select **Start > Programs > IBM Software Development Platform > IBM WebSphere Integration Developer > IBM WebSphere Integration Developer**
  - \_\_\_ b. From the Workspace Launcher window, enter **<WORKSPACE>** for the Workspace field



- \_\_\_ 3. Click on the  button on the right corner to close the Welcome page and proceed with the workbench
- \_\_\_ 4. Follow the instructions in the “**Configuring a data source**” lab with these inputs and create the data source and data base required for this lab:
  - \_\_\_ a. Data source name: **Email**
  - \_\_\_ b. JNDI name: **jdbc/Email**
  - \_\_\_ c. Database name: **EmailDB**

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

---

### Part 2: Pass through scenario

Inbound support can be broadly classified into two flows, one that involves data transformation and another without it (pass-through). In this part of the lab, you will configure the **pass through scenario** using the new External Service option from the WebSphere Integration Developer and then test the configuration with different cases.



## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

## 2.1. Configure pass through using the external service wizard

In this part of the lab you will use this new External Service feature to create and configure the Function Selector, Data Binding and other required artifacts to test the inbound pass through scenario

\_\_\_ 1. Create the module: EmailPSInboundModule

\_\_\_ a. From the Business Integration window, right-click and select **New > Module**

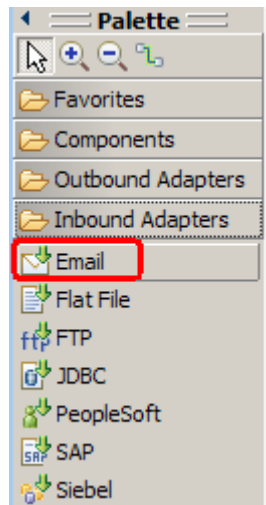
\_\_\_ b. From the New Module window, enter **EmailPSInboundModule** for the Module Name

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

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

\_\_\_ 2. To start External Service from the Palette:

\_\_\_ a. From the **Palette** on the left side of Assembly Diagram, click on **Inbound Adapters**:



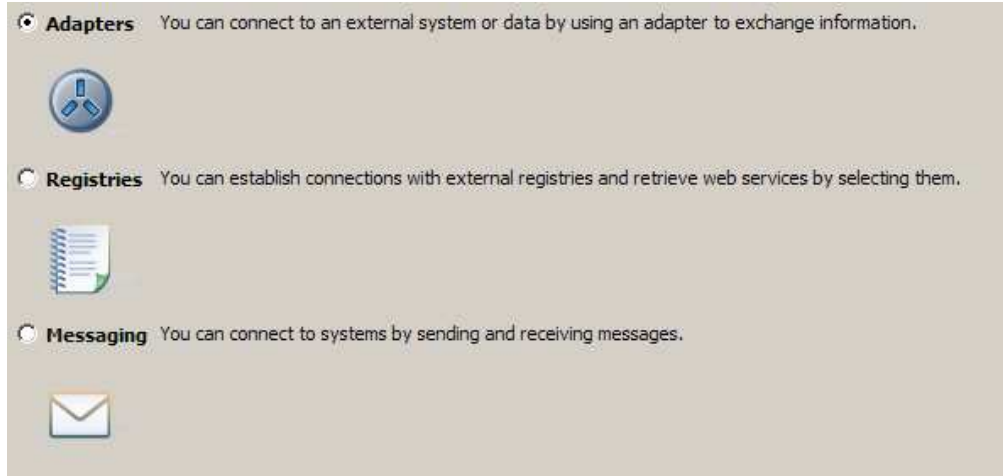
\_\_\_ b. Under Inbound Adapters, click on the **Email** and then click on the empty canvas of the assembly diagram. The New Flat File Service wizard is opened

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

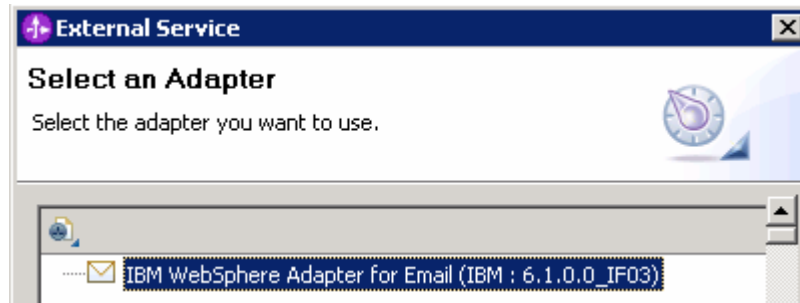
From the main menu, select **File > New > External Service**. This opens an External Service wizard that helps you obtain a service which establishes connectivity with other systems. The wizard provides three connectivity options – Adapters, Registers, and Messaging

Select the radio button next to **Adapters** and click **Next**

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE



- \_\_\_ 3. On the Select an Adapter screen, select **IBM WebSphere Adapter for Email (IBM : 6.1.0.0\_IF03)** and click **Next**



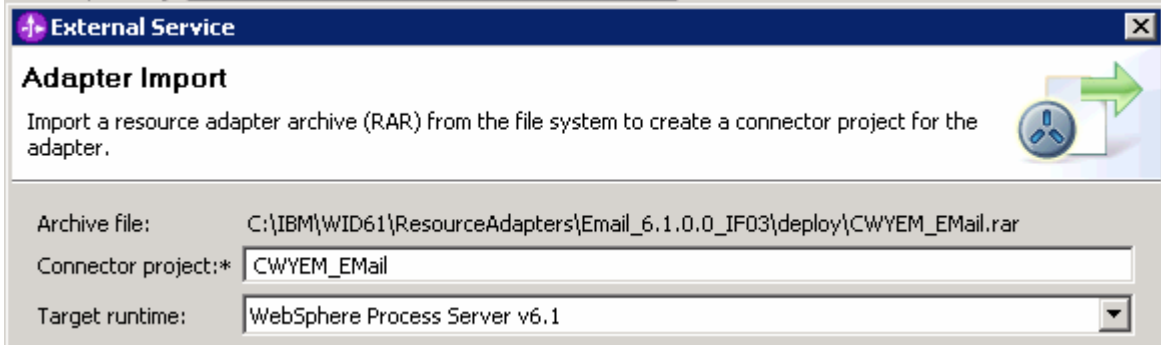
- \_\_\_ 4. Adapter Import screen:

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

- \_\_\_ a. The default Connector file is selected which is shipped along with WebSphere Integration Developer
- \_\_\_ b. Accept the default name for Connector project, **CWYEM\_Email**. You can change it to any other name, but for this lab, you can leave the default name.

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ c. For Target server, ensure that **WebSphere Process Server V6.1** is selected



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

---

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

---

---

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

---

\_\_\_ 5. Service Configuration Properties:

\_\_ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected

\_\_ b. Enter E-mail system connection information:

- 1) Host name: **<EmailServer\_HostName>** (or IP Address of the machine that has E-mail Server), for Ex: localhost
- 2) Port number: **110** (default, you should change it to the correct port number if your E-mail server is running on a different port)
- 3) Protocol: accept the default **POP3**
- 4) Not that the **Poll folder** is by default selected as **Inbox** and is not available

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

5) For **Staging directory**, click on **Browse...** and select **<STAGING>** and click **OK**

Deploy connector project: With module for use by single application

Connection properties: Use properties below

Connection properties

Staging directory: \* C:\Labfiles61\EmailInbound\StagingDir

E-mail system connection information

Host name: localhost

Port number: 110

Protocol: pop3

Poll folder or folders: Inbox

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

<< Advanced

- ▶ Event polling configuration
- ▶ Event delivery configuration
- ▶ Event persistence configuration
- ▶ Advanced properties
- ▶ Bidi properties
- ▶ Logging and tracing

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

\_\_ d. Event polling configuration: This has all the polling configuration details and for this lab, you can accept the defaults.

\_\_ e. Event delivery configuration:

- 1) **Ensure once-only event delivery:** You should check this box only if you are using data source and table name in the Event persistence configuration (below). If this property is set to true, while using in-memory capability (explained below), the adapter will log a warning message. By default this is selected and you can accept the default selection.

\_\_ f. Event persistence configuration:

- 1) Ensure that the **Auto create event table** is **checked**
- 2) Event recovery table name: **EmailPSTable**
- 3) Event recovery data source (JNDI) name: **jdbc/Email**

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

**Note:** This represents the JNDI name of the Data Source used by Event Persistence to get the JDBC database connection. The Data source must be created in the WebSphere Process Server. You should enter the data source JNDI name that you created in Step 3 of Part 1.

Event persistence configuration

Auto create event table

Event recovery table name:

Event recovery data source (JNDI) name:

User name used to connect to event data source:

Password used to connect to event data source:

Database schema name:

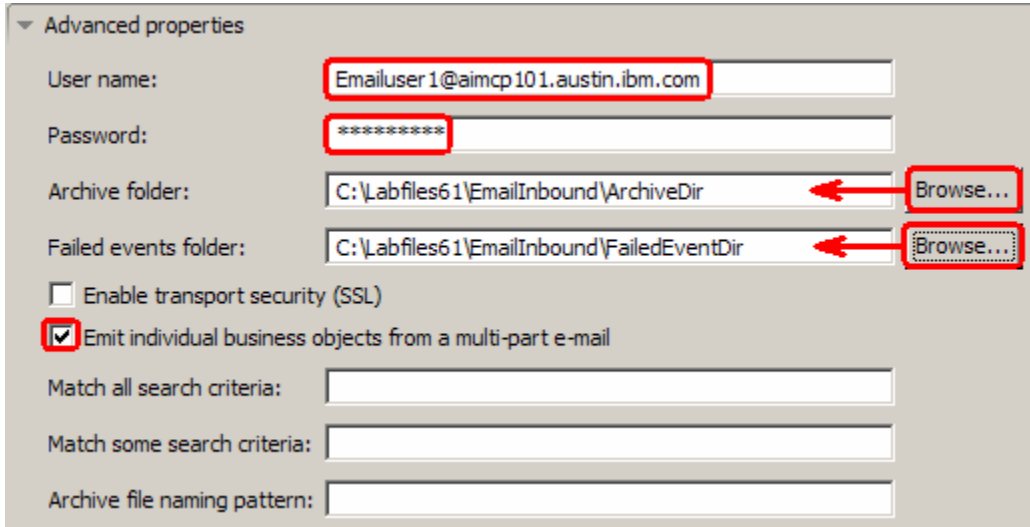
**Note:** The Event recovery data source (JNDI) name is **not mandatory** in V6.1. Now, the adapter can use **in-memory representation** of event table to store all the necessary information. Adapter uses this feature when event database information is not configured during inbound event polling. This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

Advanced properties:

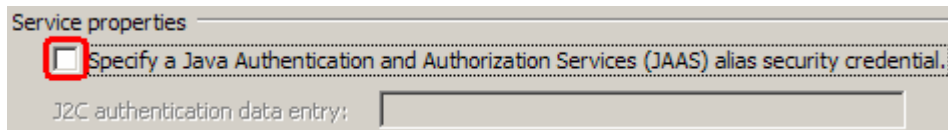
- \_\_\_ g. User name: **username using which you connect to your E-mail server** (for Ex: Emailuser1@aimcp101.austin.ibm.com)
- \_\_\_ h. Password: **password for the above user to connect to your E-mail server**
- \_\_\_ i. Archive folder: All the processed events are archived in this folder
  - 1) Click on **Browse...** next to it and select **<ARCHIVE>** and click **OK**
- \_\_\_ j. Failed event folder: All the failed events are archived in this folder
  - 1) Click on **Browse...** next to it and select **<FAILED>** and click **OK**
- \_\_\_ k. Check the box next to **Emit individual business object from a multi-part e-mail**

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

**Note:** If you select **Emit individual business object from a multi-part e-mail**, the split for individual BOs is done at the RA level and each of the BO is emitted as generic E-mail. Each part of a multipart E-mail (for ex: E-mail containing body, attachments, and file references) is considered as an individual BO and emitted using E-mail Wrapper, where **each part’s content is set in the mail content attribute**.



- \_\_\_ 6. For this lab, you are not going to use the J2C authentication. So, **uncheck** the box next to **Specify a Java Authentication and Authorization Services (JAAS) alias security credentials**.



**Function Selector Configuration:** Function Selectors are required in order to map between events generated by resource adapters, and the appropriate SCA export function name.

The E-mail Function Selector is used by the E-mail RA to get back the SCA export function name corresponding to the event type sent out by the adapter.

- \_\_\_ 7. Under Service properties, for **Function selector**, select **Use default function selector ‘EmailFunctionSelector’** from the drop down list

**Note:** Follow the steps in **Appendix: Define Function selector** to manually define a function selector.

- \_\_\_ 8. You can define data binding in two places - service level (current screen of External Service wizard) or later at the method level (Operations screen of the External Service wizard). In this lab, you will define data binding at the service level (from this screen)

- \_\_\_ a. From the dropdown menu next to Data binding, select **‘Use a data binding configuration for all operations’**



- \_\_\_ b. Click **New...** next to **Data binding configuration**. A Resource Configuration window is opened.

### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- 1) Ensure that the selected module is **EmailPSInboundModule**
- 2) For the **Name**, enter any string, for Ex: **EmailPSDB**

Module:

Namespace:

Use default namespace

Folder:

Name: \*

3) Click **Next**

- \_\_\_ c. Accept the default data binding Class Name, **com.ibm.j2ca.email.emd.runtime.EmailwrapperDataBinding**:

**Data binding**: A data binding represents the mapping between a native data format and a business object.

**Data handler**: A data handler is used by a data binding or function selector to transform data from one format to another.

**Function selector**: A function selector assigns incoming messages or requests to the correct operation on the service.

Data binding class name:

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

- \_\_\_ e. For pass through, you **do not** need to configure any data handler and business object type (because, you are using the unstructured content). So, click **Finish** on this screen

\_\_\_ f. Now the **EmailPSDB** is displayed for Data binding:

Data binding:

Data binding configuration: \*

IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ 9. Check the box next to **Change logging properties for wizard** to view the output location of the log file and the logging level. You can change the logging level using the drop down menu.
- \_\_\_ 10. Click **Next** from Connection Properties screen

Following screen is the Operations screen where you can define all your operations.

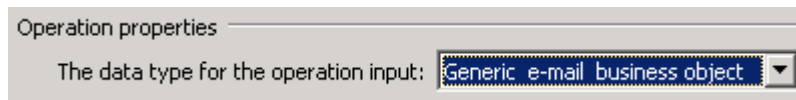
In V6.1, you can select from three different Data types for any operation:

- Generic e-mail business object
- Generic e-mail business object with business graph
- User defined type

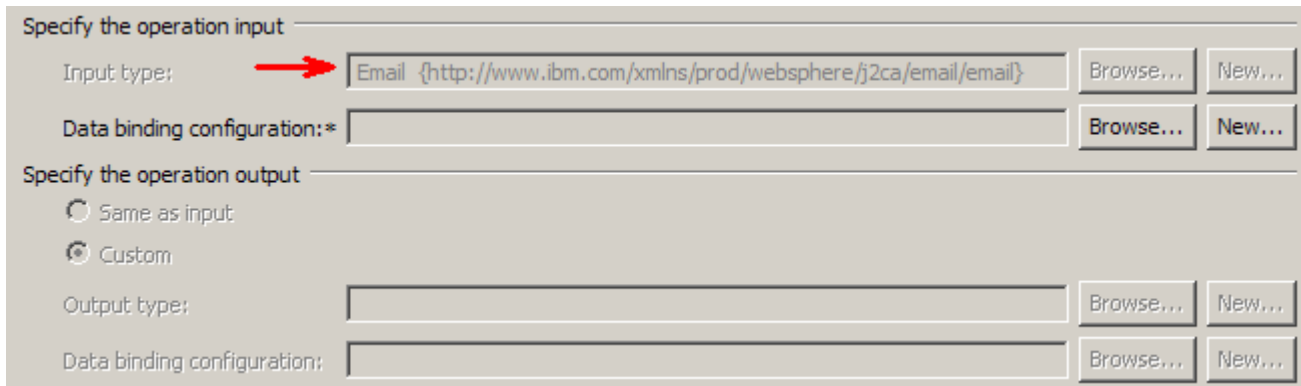
In this part of the lab, you will use the first option, **Generic e-mail business object**.

**Create emitEmailPS Operation:**

- \_\_\_ 11. From the Operations screen, click on **Add...**
  - \_\_\_ a. Add Operation window is opened. Select **Generic e-mail business object** for the Data type and click **Next**



You are now back to Operation window and because you have chosen the 'Generic e-mail business object', the Input type is populated as **Email**

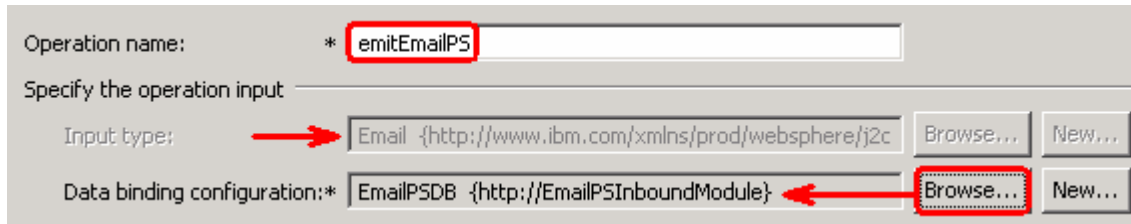


- \_\_\_ b. For Operation name, enter any name, for Ex: **emitEmailPS**
- \_\_\_ c. Note that the default Input type, **Email**
- \_\_\_ d. Define and Data Binding type:
  - 1) Click **Browse...** next to **Data binding configuration**. A Data Binding Selection window is opened.
  - 2) Select **EmailPSDB** under Matching data bindings and click **OK**



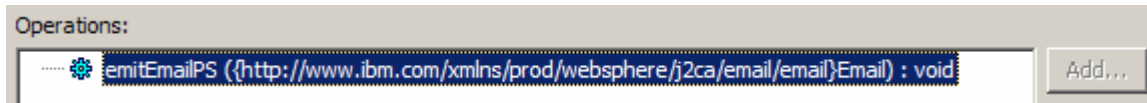
### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ e. Now the **EmailPSDB** is displayed for Data binding configuration in the Add Operation window:



\_\_ f. Click **Finish**.

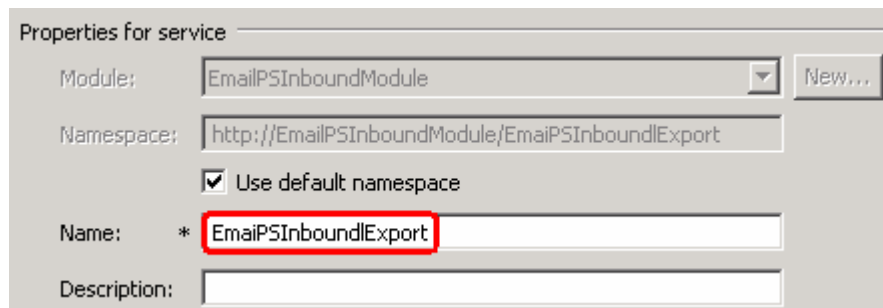
\_\_\_ 12. The above defined operation, **emitEmailPS**, is populated under Operations list



\_\_ a. Click **Next** from Operations screen

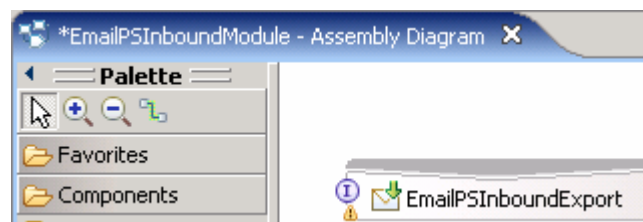
\_\_\_ 13. From Generate Artifacts screen:

\_\_ a. For **Name**, enter **EmailPSInboundExport**



\_\_ b. Click **Finish**

\_\_\_ 14. The Assembly diagram for EmailPSInboundModule is opened with an export component, EmailPSInboundExport:



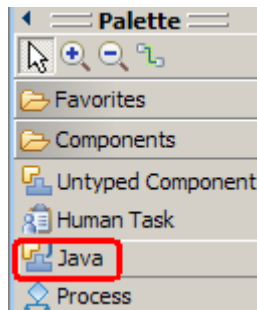
\_\_\_ 15. Save (**Ctrl + S**) changes to your assembly diagram


## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

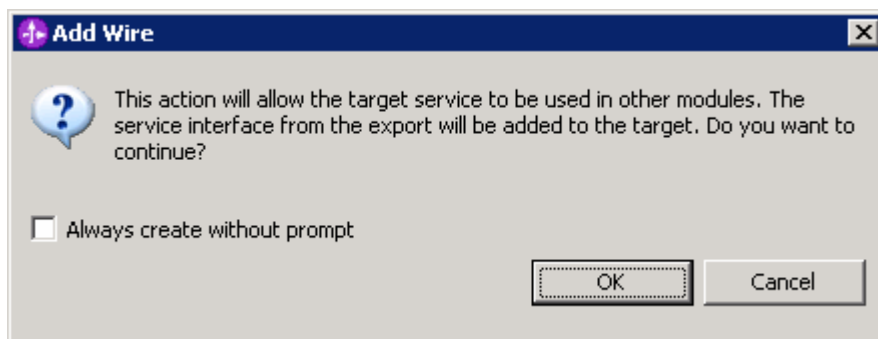
## 2.2. Add Java™ component


In this part of the lab, you will add a Java component and then wire the component to the existing export interface. Then you will continue to add a simple print Java code to the added Java component.

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



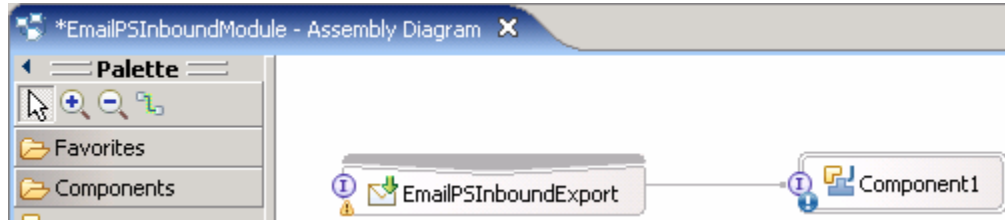
- \_\_\_ b. Click on **Java** and then click on the empty space of EmailPSInboundModule assembly diagram. This will place a new component, **Component1** on the assembly diagram.
- \_\_\_ 3. Wire the EmailPSInboundExport to the Component1
  - \_\_\_ a. Select the **wire** () icon from the Palette
  -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



- \_\_\_ d. From the top of the Palette, click on the **Selection Tool** icon () to get back to the normal cursor mode
- \_\_\_ e. Right-click on the empty space of the Assembly diagram and select **Arrange Contents Automatically** from the pop-up menu

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

Your assembly diagram for EmailPSInboundModule will look like this:



- \_\_ f. Right-click on **Component1** and select **Generate Implementation** from the pop-up menu
- \_\_ g. On the **Generate Implementation** panel, select **default package**, and click **OK**
- \_\_ h. **Component1Impl.java** is opened in Assembly editor. Scroll down to the method **emitEmailPS(DataObject emitEmailPSInput)** that needs to be implemented and add this code under that method:

```
public void emitEmailPS(DataObject emitEmailPSInput) {
    //TODO Needs to be implemented.

    System.out.println("*****Reached Endpoint: EmailPS*****");

    DataObject unstructured =
    emitEmailPSInput.getDataObject("mailContent");

    System.out.println("*****Email Pass through content*****");

    String astext = unstructured.getString("AsText");
    System.out.println("As Text content-----> "+astext);
}
```

---

**Note:** The code is also available at <EMAILFILES>\ PSInboundJavaCode.txt

---

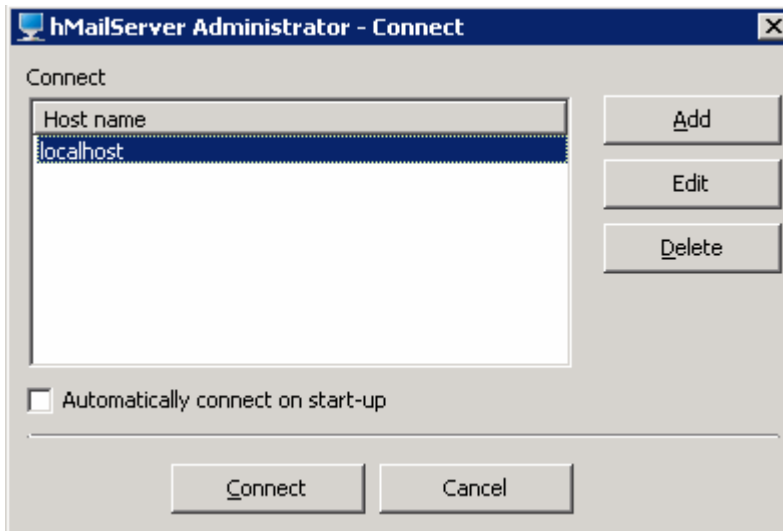
- \_\_ i. Save (**Ctrl + S**) and close Component1Impl.java
- \_\_ j. Save (**Ctrl + S**) and close Assembly diagram: EmailPSInboundModule

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

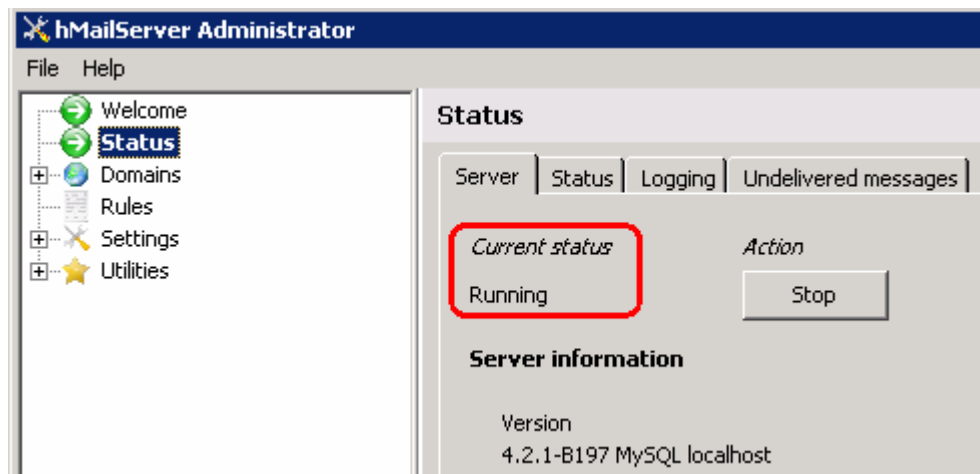
## 2.3. Test pass through scenario

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

- \_\_\_ 1. Start your e-mail server (if not started already)
  - \_\_\_ a. Select **Start > hMailServer > hMailServerAdministrator**
  - \_\_\_ b. From the hMailServer Administrator – Connect window, ensure that **localhost** is selected and click on **Connect**



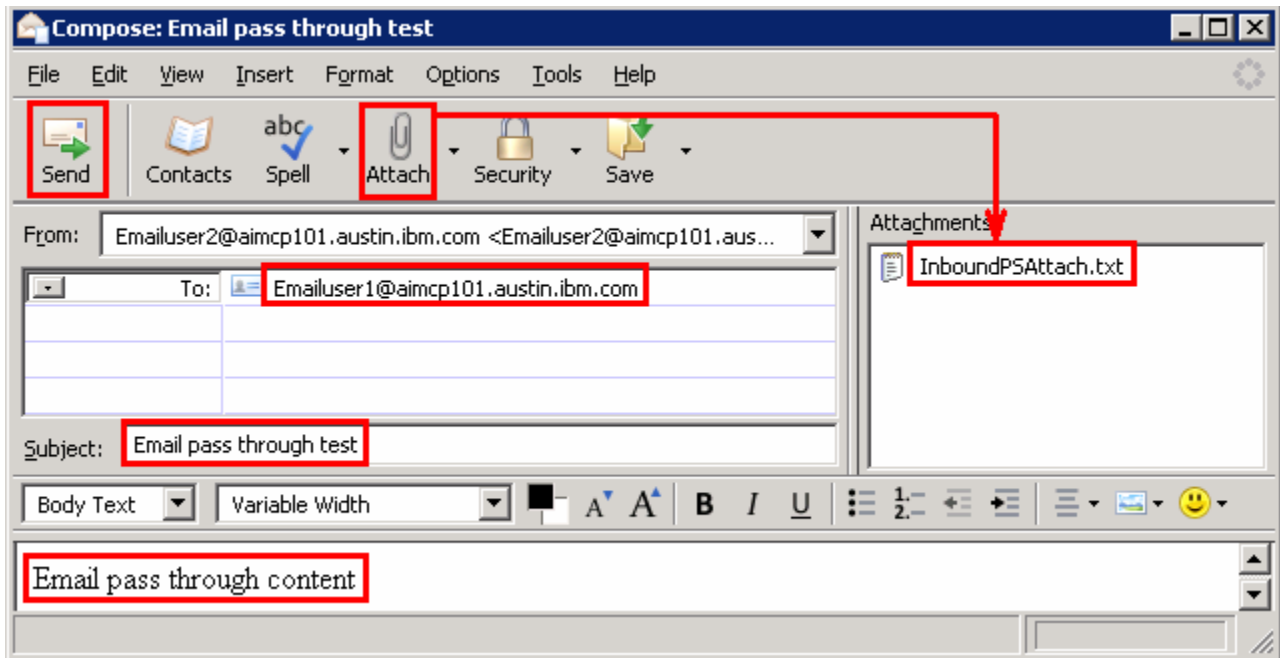
- \_\_\_ c. hMailServer Administrator window is opened and the Current status should show **Running**



- \_\_\_ 2. Add the project to the WebSphere Test Environment server
  - \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
  - \_\_\_ b. From the Add and Remove Projects window, select **EmailPSInboundModuleApp** under Available projects panel and click **Add >**

### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ c. You will now see the **EmailPSInboundModuleApp** added to the **Configured projects**
- \_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server is started in Debug mode if it is not already started before
- \_\_\_ 3. Test the adapter application: send a pass through event e-mail with an attachment from the Mozilla Thunderbird client,
  - \_\_\_ a. From the Mozilla Thunderbird, select **Emailuser2** and then click on **Write** icon at the top of the window. A compose window is opened
  - \_\_\_ b. To: [Emailuser1@aimcp101.austin.ibm.com](mailto:Emailuser1@aimcp101.austin.ibm.com)
  - \_\_\_ c. Subject: Email pass through test
  - \_\_\_ d. Body (Content): Email pass through content
  - \_\_\_ e. Click on **Attach** from the top of Compose window and select **<EMAILFILES>InboundPSAttach.txt**. Note the content in this txt file so that you can verify in the coming steps.



- \_\_\_ f. Click **Send** from the top menu
- \_\_\_ g. Select text only option if you get any pop-up window
- \_\_\_ h. You can see the E-mail in the inbox of Emailuser1, but it is deleted quickly by the adapter after polling.
- \_\_\_ 4. Verify the results:
  - \_\_\_ a. Because the adapter is running, the event E-mail will pass through the **emitEmailPS** method and you should see this message in the console (or SystemOut.log):

IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

**Note:** Since you have chosen the option **'Emit individual business object from a multi-part e-mail'**, the content and the attachment are considered as individual BOs and hence it reached the end point twice once for the content and then for the attachment. And **each part's content is set in the mail content attribute**

```
[1/7/08 12:00:13:625 CST] 0000092 SystemOut      O *****Reached Endpoint:
EmailPS*****

[1/7/08 12:00:13:625 CST] 0000092 SystemOut      O *****Email Pass
through content*****

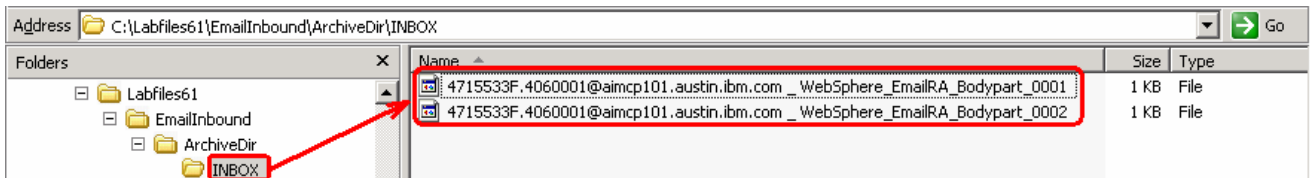
[1/7/08 12:00:13:625 CST] 0000092 SystemOut      O As Text content-----> Email
pass through content

[1/7/08 12:00:13:640 CST] 000008d SystemOut      O *****Reached Endpoint:
EmailPS*****

[1/7/08 12:00:13:640 CST] 000008d SystemOut      O *****Email Pass
through content*****

[1/7/08 12:00:13:640 CST] 000008d SystemOut      O As Text content-----> Email
inbound pass through attachment content
```

\_\_ b. Check the **ArchiveDir** which should contain a folder **INBOX** (created by the adapter) and if you open that **INBOX** folder, you will see two new files with the name ending in **Bodypart\_0001** and **Bodypart\_0002**.



\_\_ c. Open the file name ending with **Bodypart\_0001** under **INBOX** and observe that it is an archive of the content:



## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ d. Open the file name ending with **Bodypart\_0002** under **INBOX** and observe that it is an archive of the attachment:

```
47826A60.1060104@aimcp101.austin.ibm.com_WebSphere_EmailRA_Bodypart_0002 - Notepad
File Edit Format View Help
Return-Path: <Emailuser2@aimcp101.austin.ibm.com>
Received: from [9.3.75.101] ([9.3.75.101])
    by localhost
    with hMailServer ; Mon, 7 Jan 2008 12:07:28 -0600
Message-ID: <47826A60.1060104@aimcp101.austin.ibm.com>
Date: Mon, 07 Jan 2008 12:07:28 -0600
From: "Emailuser2@aimcp101.austin.ibm.com" <Emailuser2@aimcp101.austin.ibm.com>
User-Agent: Thunderbird 2.0.0.9 (windows/20071031)
MIME-Version: 1.0
To: Emailuser1@aimcp101.austin.ibm.com
Subject: Email pass through test
Content-Type: text/plain; name=InboundPSAttach.txt; charset=us-ascii
Content-Transfer-Encoding: 7bit

Email inbound pass through attachment content
```

- \_\_\_ 5. Restore the Sever Configuration
- \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
  - \_\_\_ b. Select **EmailPSInboundModuleApp** 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: Content specific (non-pass through) scenario**

Of the two flows, you have just tested the pass through scenario that does not involve data transformation. In this part of the lab, you will configure the **non-pass through scenario** using the new External Service option from the WebSphere Integration Developer and then test the configuration with different cases.



## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

### 3.1. Configure non-pass through using the external service wizard

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

\_\_\_ 1. Create the module: EmailNonPSInboundModule

\_\_\_ a. From the Business Integration window, right-click and select **New > Module**

\_\_\_ b. From the New Module window, enter **EmailNonPSInboundModule** for the Module Name

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

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

\_\_\_ 2. Import required business objects

\_\_\_ a. Expand EmailNonPSInboundModule (if not already expanded), right-click on **Data Types** and select **Import...** from the pop-up menu

\_\_\_ b. From the Import window, expand **General** and select **File System** and then click **Next**

\_\_\_ c. Enter From directory

1) Click on **Browse...** next to **From directory**

2) From the Import from directory window, select **<EMAILFILES>** and click **OK**

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

\_\_\_ d. Select the box next to **Wbiaddress.xsd** and **Wbiphone.xsd**

\_\_\_ e. Ensure that the **EmailNonPSInboundModule** is selected for Into folder

\_\_\_ f. Click **Finish** from the Import window

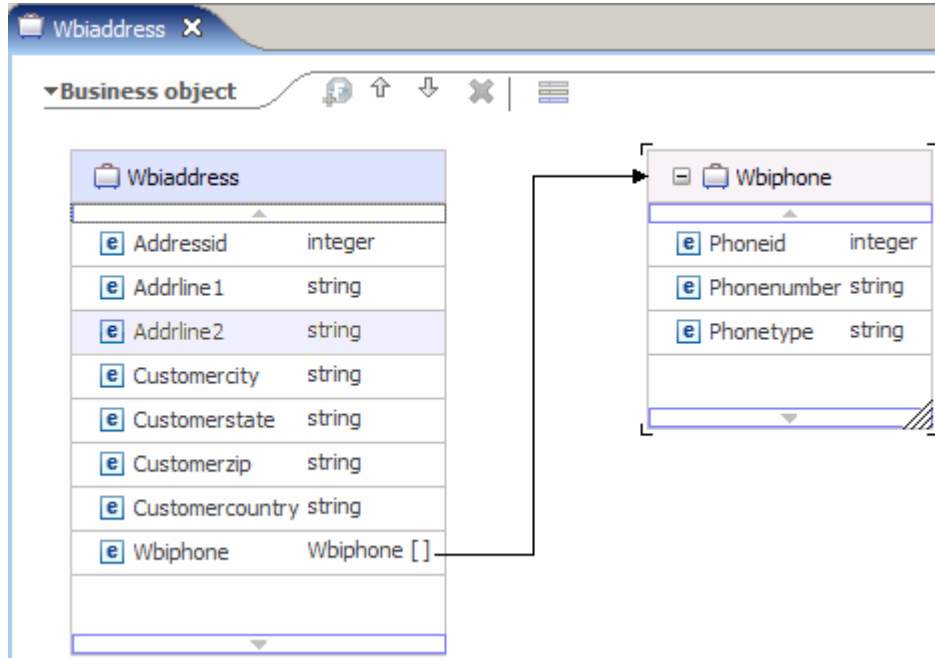
The Business Integration window is updated with the imported business objects.

\_\_\_ 3. Review the imported business objects:

\_\_\_ a. Expand **EmailNonPSInboundModule > Data Types** and you will now see a new data type **Wbiaddress** under it.

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ b. Double-click on **Wbiaddress** to open it in assembly editor and then expand Wbiphone from assembly editor to view its fields



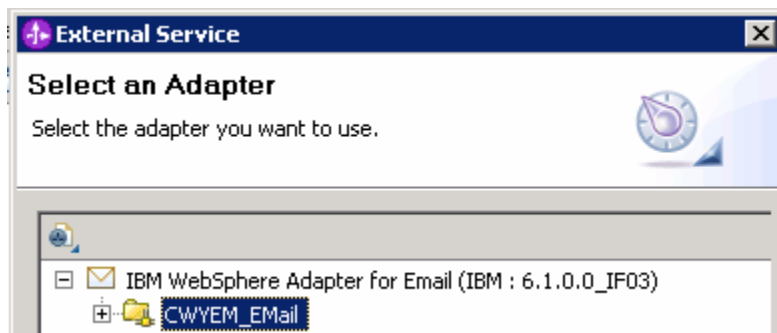
- \_\_\_ 4. After reviewing, close the Customer business object from the Assembly editor
- \_\_\_ 5. To start External Service from the Palette:
  - \_\_\_ a. From the **Palette** on the left side of Assembly Diagram, click on **Inbound Adapters**:
  - \_\_\_ b. Under Inbound Adapters, click on the **Email** and then click on the empty canvas of the assembly diagram. The New Flat File Service wizard is opened

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

From the main menu, select **File > New > External Service**. This opens an External Service wizard that helps you obtain a service which establishes connectivity with other systems. The wizard provides three connectivity options – Adapters, Registers, and Messaging

Select the radio button next to **Adapters** and click **Next**

- \_\_\_ 6. On the Select an Adapter screen, select **IBM WebSphere Adapter for Email (IBM : 6.1.0.0\_IF03)** and click **Next**



## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

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

\_\_\_ 7. Service Configuration Properties:

\_\_\_ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected

\_\_\_ b. Enter E-mail system connection information:

- 1) Host name: **<EmailServer\_HostName>** (or IP Address of the machine that has E-mail Server), for Ex: localhost
- 2) Port number: **110** (default, you should change it to the correct port number if your E-mail server is running on a different port)
- 3) Protocol: accept the default **POP3**
- 4) Not that the **Poll folder** is by default selected as **Inbox** and is not available
- 5) For **Staging directory**, click on **Browse...** and select **<STAGING>** and click **OK**

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

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

\_\_\_ d. Event delivery configuration:

- 1) **Ensure once-only event delivery:** You should check this box only if you are using data source and table name in the Event persistence configuration (below). If this property is set to true, while using in-memory capability (explained below), the adapter will log a warning message. By default this is selected and you can accept the default selection.

\_\_\_ e. Event persistence configuration:

- 1) Ensure that the **Auto create event table** is **checked**
- 2) Event recovery table name: **EmailNonPSTable**

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

3) Event recovery data source (JNDI) name: **jdbc/Email**

**Note:** This represents the JNDI name of the Data Source used by Event Persistence to get the JDBC database connection. The Data source must be created in the WebSphere Process Server. You should enter the data source JNDI name that you created in Step 3 of Part 1.

Event persistence configuration

Auto create event table

Event recovery table name:

Event recovery data source (JNDI) name:

User name used to connect to event data source:

Password used to connect to event data source:

Database schema name:

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

Advanced properties:

- \_\_ f. User name: **username using which you connect to your E-mail server** (for Ex: Emailuser1@aimcp101.austin.ibm.com)
- \_\_ g. Password: **password for the above user to connect to your E-mail server**
- \_\_ h. Archive folder: All the processed events are archived in this folder
  - 1) Click on **Browse...** next to it and select **<ARCHIVE>** and click **OK**
- \_\_ i. Failed event folder: All the failed events are archived in this folder
  - 1) Click on **Browse...** next to it and select **<FAILED>** and click **OK**

Advanced properties

User name:

Password:

Archive folder:

Failed events folder:

Enable transport security (SSL)

Emit individual business objects from a multi-part e-mail

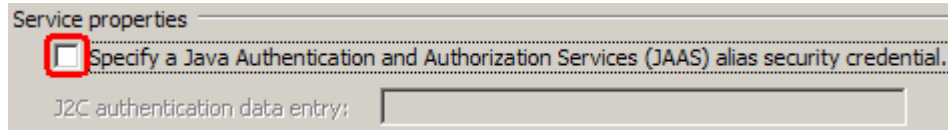
Match all search criteria:

Match some search criteria:

Archive file naming pattern:

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ 8. For this lab, you are not going to use the J2C authentication. So, **uncheck** the box next to **Specify a Java Authentication and Authorization Services (JAAS) alias security credentials**.



**Function Selector Configuration:**

- \_\_\_ 9. Under Service properties, for **Function selector**, select **Use default function selector 'EmailFunctionSelector'** from the drop down list

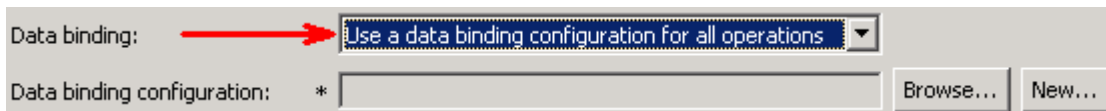
---

**Note:** Follow the steps in **Appendix: Define Function selector** to manually define a function selector.

---

- \_\_\_ 10. You can define data binding in two places - service level (current screen of External Service wizard) or later at the method level (Operations screen of the External Service wizard). In this lab, you will define data binding at the service level (from this screen)

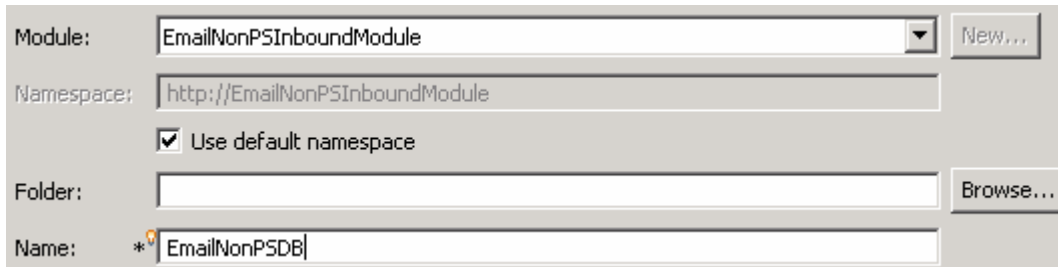
- \_\_\_ a. From the dropdown menu next to Data binding, select **'Use a data binding configuration for all operations'**



- \_\_\_ 11. Click **New...** next to **Data binding configuration**. A Resource Configuration window is opened.

- \_\_\_ a. Ensure that the selected module is **EmailNonPSInboundModule**

- 1) For the **Name**, enter **EmailNonPSDB**



- 2) Click **Next**

- \_\_\_ b. Accept the default data binding Class Name, **com.ibm.j2ca.email.emd.runtime.EmailwrapperDataBinding** and click **Next**

- \_\_\_ c. Click on **Add...** from Data binding properties table. Add/Edit window is opened

- \_\_\_ d. For **Binding type**, ensure that **DataHandler** is selected

- \_\_\_ e. Click on **Browse...** next to **Business object type**

- 1) From Data Type Selection window, select **Wbiaddress** and click **OK**

You are now back to the Add/Edit screen and the above defined business object, Wbiaddress, is populated in this screen:

### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ f. For **Mime type**, select **text/xml** from the drop down list

Binding type: DataHandler

Business object type: \* Wbiaddress {http://www.ibm.com/xmlns/prod/websphere/j2ca/email/wbiaddress} Browse... New...

Mime type: \* text/xml

Configured data handler:\* Browse... New...

Configured data binding: Browse... New...

#### Data Handler Configuration:

\_\_ g. Click on **New...** next to **Configured data handler**. A new Resource Configuration window is opened for you to define the data handler

\_\_ h. Ensure that the module selected is **EmailNonPSInboundModule** and enter **EmailNonPSDH** for the Name of the data handler that is going to be created.

Module: EmailNonPSInboundModule New...

Namespace: http://EmailNonPSInboundModule

Use default namespace

Folder: Browse...

Name: \* EmailNonPSDH

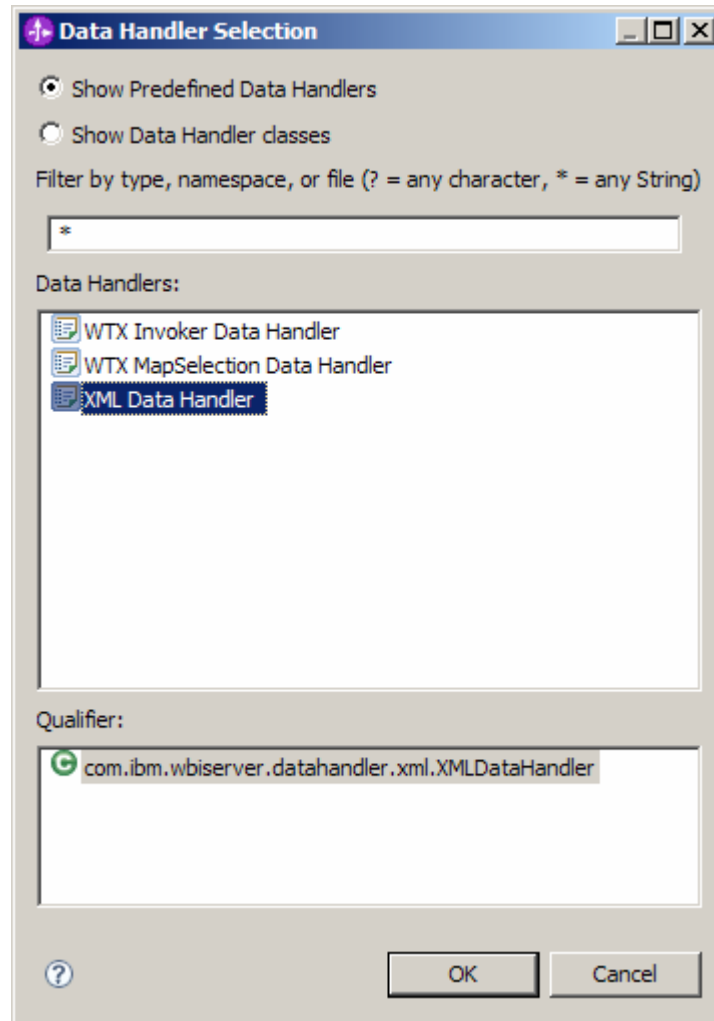
\_\_ i. Click **Next**

By default, the radio button next to **Data Handler** is selected

\_\_ j. Click on **Browse...** next to Data handler class name. Data Handler Selection window is opened.

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

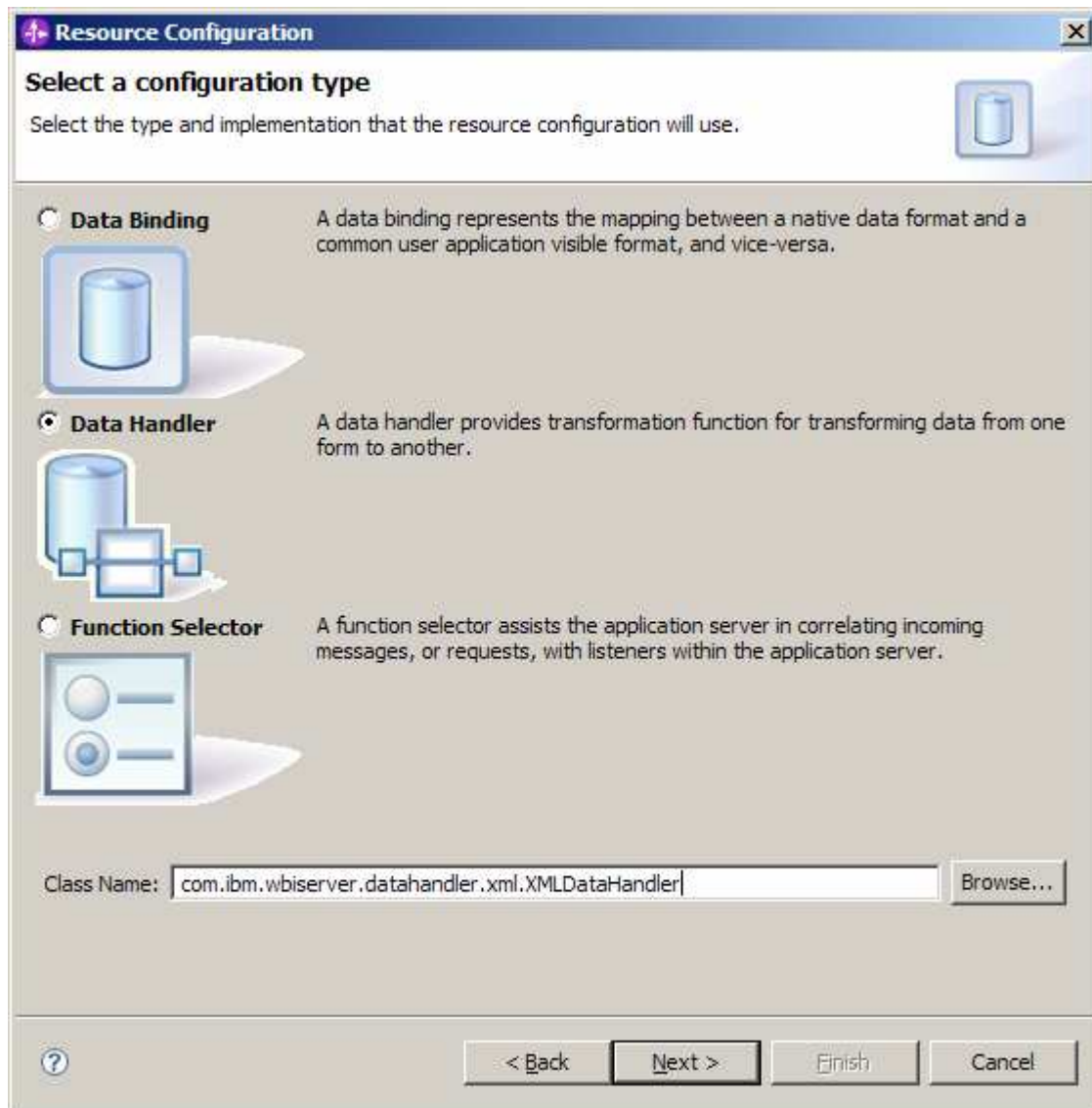
\_\_ k. Select **XML Data Handler** from the Data Handlers list



\_\_ l. Click **OK**

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_ m. You are now back to Resource Configuration window and the above selected data handler, **com.ibm.wbiserver.datahandler.xml.XMLDataHandler**, is displayed as the Class Name of Data Handler:



- \_\_ n. Click **Next**

- \_\_ o. Accept the default selection 'UTF-8' for encoding and click **Finish**



IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ p. You are now done with defining the data handler and back to Add/Edit properties screen. The Data handler configuration name, **EmailNonPSDH** is populated in this screen

\_\_\_ q. Click **Finish** from the Add/Edit properties screen

\_\_\_ r. You are now back to the Data binding properties screen with a new entry

Data binding properties:						
Select DataBindin...	Binding type	Business object type	Mime type	Configured data handler	Configured data binding	Add...
Select DataBindin...	DataHandler	{http://www.ibm.com/xmlns/prod/websphere/j2ca/email/wbiaddress}Wbiaddress	text/xml	{http://EmailNonPSInboundModule}EmailNonPSDH		Edit...

\_\_\_ s. Click **Finish** from Data binding properties screen

\_\_\_ t. The **EmailNonPSDB** is displayed for Data binding configuration:

\_\_\_ 12. Check the box next to **Change logging properties for wizard** to view the output location of the log file and the logging level. You can change the logging level using the drop down menu

\_\_\_ a. Click **Next**

Following screen is the Operations screen where you can define all your operations.

In V6.1, you can select from three different Data types for any operation:

- Generic e-mail business object
- Generic e-mail business object with business graph
- User defined type

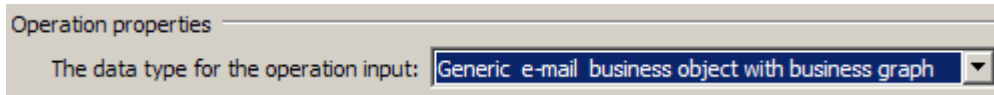
In this part of the lab, you will use the first option, **Generic e-mail business object with business graph**

**Create emitEmailNonPS Operation:**

IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

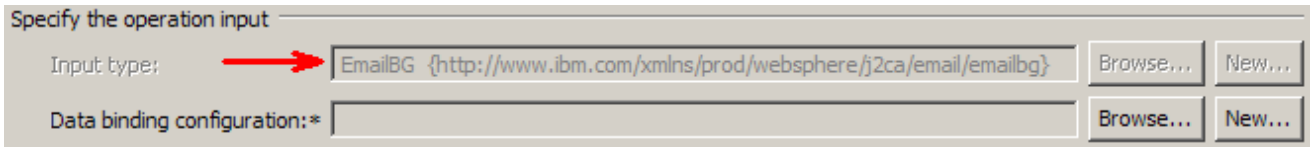
\_\_\_ 13. Click on **Add...** to open Add Operation window

\_\_\_ a. For **Data type for the operation input**, select **Generic e-mail business object with business graph** from the drop down list



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

The Input type is populated based on the selection of the Data type for the operation in the previous step. Since you have chosen Generic e-mail business object with business graph, the Input type is **EmailBG**



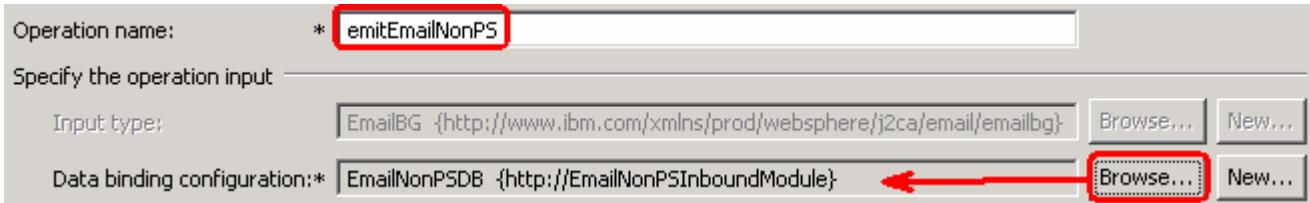
\_\_\_ c. For Operation name, enter **emitEmailNonPS**

Define Data Binding configuration:

\_\_\_ d. Under **Specify the operation input**, click on **Browse...** next to **Data binding configuration**. A Data Binding Selection window is opened.

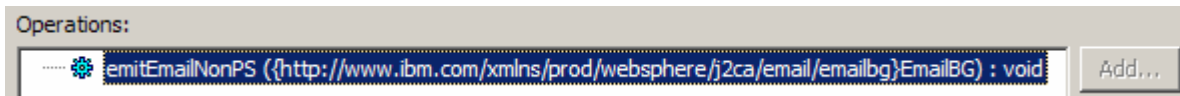
\_\_\_ e. Select **EmailNonPSDB** under Matching data bindings and click **OK**

\_\_\_ f. Now the **EmailNonPSDB** is displayed for Data binding configuration in the Add Operation window:



\_\_\_ g. Click **Finish**.

\_\_\_ 14. The above defined operation, **emitEmailNonPS**, is populated under Operations list:

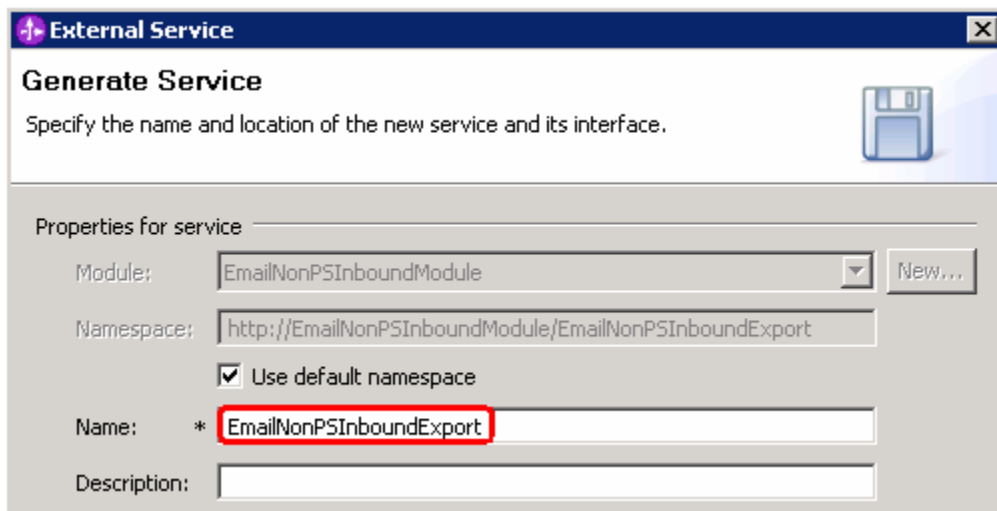


\_\_\_ a. Click **Next** from Operations screen

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

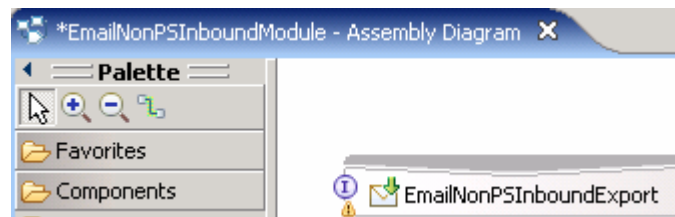
\_\_\_ 15. From Generate Artifacts screen:

\_\_\_ a. For Name, enter **EmailNonPSInboundExport**



\_\_\_ b. Click **Finish**

\_\_\_ 16. The Assembly diagram for EmailNonPSInboundModule is opened with an Export component, EmailNonPSInboundExport:





\_\_\_ a. Save (**Ctrl + S**) changes to your assembly diagram

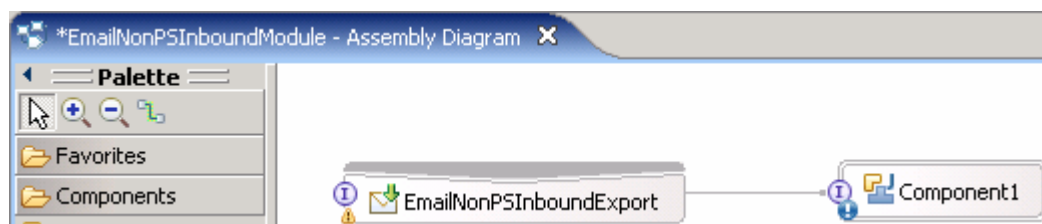
## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

## 3.2. Add Java component

In this part of the lab, you will add a Java component and then wire the component to the existing export interface. Then you will continue to add a simple print Java code to the added Java component.

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

Your assembly diagram for EmailNonPSInboundModule will look like this:



- \_\_\_ 4. Generate implementation for Component1
  - \_\_\_ a. Right-click on **Component1** and select **Generate Implementation** from the pop-up menu
  - \_\_\_ b. On the **Generate Implementation** panel, select **default package**, and click **OK**
  - \_\_\_ c. **Component1Impl.java** is opened in Assembly editor.
    - 1) From the top of the Java implementation, under imports, type this to add one more import:
 

```
import java.util.*;
```
    - 2) Scroll down to the method **emitEmailNonPS(DataObject emitEmailNonPSInput)** that needs to be implemented and add this code under that method:

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

```

System.out.println("*****ENDPOINT
emitNonPSEmail*****");

DataObject EMail = emitEmailNonPSInput.getDataObject("Email");
DataObject content = (DataObject) EMail.get("mailContent");
System.out.println("*****EMail Content*****");
String Addressid = content.getString("Addressid");
System.out.println("Address ID-----> "+Addressid);
String AddrLine1 = content.getString("Addrline1");
System.out.println("Address Line1--> "+AddrLine1);
String AddrLine2 = content.getString("Addrline2");
System.out.println("Address Line2--> "+AddrLine2);
String CustomerCity = content.getString("Customercity");
System.out.println("Customer City-----> "+CustomerCity);
String CustomerState = content.getString("Customerstate");
System.out.println("Customer State-----> "+CustomerState);
String CustomerZip = content.getString("Customerzip");
System.out.println("Customer Zip-----> "+CustomerZip);
String CustomerCountry = content.getString("Customercountry");
System.out.println("Customer Country-----> "+CustomerCountry);

//Get all the attachments as an ArrayList and iterate through
the list

List list = EMail.getList("mailAttachments");
String attachName = null;
DataObject tempSDO=null;
//Extract each attachment
for (int i = 0; i<list.size(); i++){
    Object attachmentObject = list.get(i);
    tempSDO = (DataObject)list.get(i);
//Attachment name and content
    attachName = tempSDO.getString("attachmentName");
    System.out.println("***ATTACHMENT NAME: "+attachName
+ "***");

```

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

```
DataObject bo = (DataObject) tempSDO.get("appBOData");

System.out.println("*****Attachment
Content*****");

String PhoneID = bo.getString("Phoneid");

System.out.println("Phone ID-----> "+PhoneID);

String PhoneNumber = bo.getString("Phonenumber");

System.out.println("Phone Number--> "+PhoneNumber);

String PhoneType = bo.getString("Phonetype");

System.out.println("Phont Type--> "+PhoneType);
}
```

---

**Note:** The Java code is also available at <EMAILFILES>\ NonPSInboundJavaCode.txt

---

- \_\_ d. Save (**Ctrl + S**) and close Component1Impl.java
- \_\_ e. Save (**Ctrl + S**) and close Assembly diagram: EmailNonPSInboundModule

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

### 3.3. Test non pass through scenario

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

- \_\_\_ 1. Start your e-mail server (if not started already)
  - \_\_\_ a. Select **Start > hMailServer >hMailServerAdministrator**
  - \_\_\_ b. From the hMailServer Administrator – Connect window, ensure that **localhost** is selected and click on **Connect**
  - \_\_\_ c. hMailServer Administrator window is opened and the Current status should show **Running**
- \_\_\_ 2. Add the project to the WebSphere Test Environment server
  - \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
  - \_\_\_ b. From the Add and Remove Projects window, select **EmailNonPSInboundModuleApp** under Available projects panel and click **Add >**
  - \_\_\_ c. You will now see the **EmailNonPSInboundModuleApp** added to the **Configured projects**
  - \_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server will start in Debug mode if it is not already started before
- \_\_\_ 3. Review and modify the batch file to generate required event E-mails
  - \_\_\_ a. From windows explorer, browse to **<EMAILFILES>EmailEventCreator**

---

**Note:** You can find EmailEventsCreator.zip at <EMAILRAR\_FILE>\Samples also which shipped with WebSphere Integration Developer. You will need to extract it and make modifications to the batch file inside the archive file. For your convenience, you can use the files under **<EMAILFILES>EmailEventCreator** which are already modified for this lab.

---

- \_\_\_ b. Review the batch file, createEvents.bat, under EmailEventsCreator folder:

Structure of **createEvents.bat** file:

```
<host> <port> <number of events to be created> <from> <to> <cc> <bcc> <subject> <content>
<path for event file name> <Boolean for pass through/non pass through>
```

Where, Path for event file name - is the path on the local folder where the event file attachment has been placed.

Boolean for pass-through/non-pass-through – has true or false values. **True** indicates pass-through and **False** indicates non-pass-through.

- \_\_\_ c. Review **createEvents\_sample\_nonpassthrough.bat** file under EmailEventsCreator folder. This file, for your convenience, is created by providing the required inputs specified in the createEvents.bat. This file is used to generate the required E-mail events:

```
java -cp .\;\mail.jar;\activation.jar EmailEventCreatorNew "localhost" 25 1
"Emailuser2@aimcp101.austin.ibm.com" Emailuser1@aimcp101.austin.ibm.com
Emailuser3@aimcp101.austin.ibm.com Emailuser4@aimcp101.austin.ibm.com "Test Email"
```

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

```
inbound Non-pass through" "C:\Labfiles61\EmailFiles\WbiAddress.xml"  
"C:\Labfiles61\EmailFiles\WbiPhone.xml" "false"
```

---

**Note:** This will generate 1 event E-mail from [Emailuserser2@aimcp101.austin.ibm.com](mailto:Emailuserser2@aimcp101.austin.ibm.com) on the localhost mail server, to the inbox of [Emailuserser1@aimcp101.austin.ibm.com](mailto:Emailuserser1@aimcp101.austin.ibm.com) with the file **WbiPhone.xml** as an attachment and the content of **WbiAddress.xml** file as the body of the E-mail. **You should change the fields according to your E-mail server settings and file locations.**

---

- \_\_\_ 4. Run the batch file to generate email
  - \_\_\_ a. Open a command prompt and change the directory to **<EMAILFILES>EmailEventCreator**



## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ b. Enter `createEvents_sample_nonpassthrough.bat`

```
MAIL FROM:<Emailuser2@aimcp101.austin.ibm.com>
250 OK
RCPT TO:<Emailuser1@aimcp101.austin.ibm.com>
250 OK
RCPT TO:<Emailuser3@aimcp101.austin.ibm.com>
250 OK
RCPT TO:<Emailuser4@aimcp101.austin.ibm.com>
250 OK
DEBUG SMTP: Verified Addresses
DEBUG SMTP: Emailuser1@aimcp101.austin.ibm.com
DEBUG SMTP: Emailuser3@aimcp101.austin.ibm.com
DEBUG SMTP: Emailuser4@aimcp101.austin.ibm.com
DATA
354 OK, send.
Message-ID: <1762786195.1201033529625.JavaMail.Administrator@localhost>
Date: Tue, 22 Jan 2008 14:25:29 -0600 (CST)
From: Emailuser2@aimcp101.austin.ibm.com
To: Emailuser1@aimcp101.austin.ibm.com
Subject: Test Email inbound Non-pass through
Cc: Emailuser3@aimcp101.austin.ibm.com
Mime-Version: 1.0
Content-Type: multipart/mixed;
    boundary="-----_Part_0_1720073107.1201033529500"

-----=_Part_0_1720073107.1201033529500
Content-Type: text/xml; charset=us-ascii
Content-Transfer-Encoding: 7bit

<?xml version="1.0" encoding="UTF-8"?>
<wbiaddress:wbiaddress xsi:type="wbiaddress:wbiaddress" xmlns:xsi="http://www.w3
.org/2001/XMLSchema-instance" xmlns:wbiaddress="http://www.ibm.com/xmlns/prod/we
bsphere/j2ca/email/wbiaddress">
  <Addressid>98765</Addressid>
  <Addrline1>11501</Addrline1>
  <Addrline2>Burnet Rd</Addrline2>
  <Customercity>Austin</Customercity>
  <Customerstate>TX</Customerstate>
  <Customerzip>78758</Customerzip>
  <Customercountry>USA</Customercountry>
</wbiaddress:wbiaddress>

-----=_Part_0_1720073107.1201033529500
Content-Type: text/xml; name=sample.xml; charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename=sample.xml

<?xml version="1.0" encoding="UTF-8"?>
<wbiphone:wbiaddress xsi:type="wbiphone:wbiaddress" xmlns:xsi="http://www.w3.org/200
1/XMLSchema-instance" xmlns:wbiphone="http://www.ibm.com/xmlns/prod/websphere/j2
ca/email/wbiphone">
  <Phoneid>12345</Phoneid>
  <Phonenumber>5128380000</Phonenumber>
  <Phonetyp>Office</Phonetyp>
</wbiphone:wbiaddress>

-----=_Part_0_1720073107.1201033529500--
```

\_\_ c. You can see the E-mail in the inbox of Emailuser1, but it is deleted quickly by the adapter after polling.

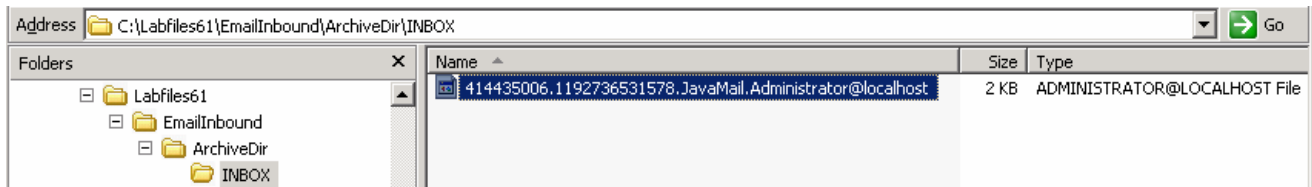
IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ 5. Verify the results:

- \_\_\_ a. Because the adapter is running, the event E-mail will pass through the **emitEmailNonPS** method and you should see these messages in the console (or SystemOut.log):

```
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O *****ENDPOINT emitNonPSEmail*****
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O *****Email Content*****
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Address ID----> 98765
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Address Line1--> 11501
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Address Line2--> Burnet Rd
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Customer City----> Austin
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Customer State----> TX
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Customer Zip----> 78758
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Customer Country----> USA
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O ***ATTACHMENT NAME: sample.xml***
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O *****Attachment Content*****
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Phone ID----> 12345
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Phone Number--> 5128380000
[1/22/08 14:25:33:531 CST] 00000089 SystemOut      O Phont Type--> Office
```

- \_\_\_ b. Check the **<ARCHIVE>** which should contain a folder INBOX (created by the adapter) and if you open that **INBOX** folder, you will see a new file with the name ending in your host name.



- \_\_\_ c. You can open the file and verify the same content that the E-mail was created with.

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

- \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
- \_\_\_ b. Select **EmailNonPSInboundModuleApp** 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 4: User defined (Fixed structure e-mail) scenario

User defined scenario is also a non-pass through but you can specify your own business object types using this option. In this part of the lab, you will configure **User defined scenario** using the new External Service option from the WebSphere Integration Developer and then test the configuration with different cases

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

## 4.1. Configure fixed structure using the external service wizard

In this part of the lab you will use this new External Service feature to create and configure the Function Selector, Data Binding and other required artifacts to test the inbound pass through scenario

\_\_\_ 1. Create the module: **EmailFixedInboundModule**

\_\_\_ a. From the Business Integration window, right-click and select **New > Module**

\_\_\_ b. From the New Module window, enter **EmailFixedInboundModule** for the Module Name

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

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

\_\_\_ 2. Import required business objects

\_\_\_ a. Expand **EmailFixedInboundModule** (if not already expanded), right-click on **Data Types** and select **Import...** from the pop-up menu

\_\_\_ b. From the Import window, expand **General** and select **File System** and then click **Next**

\_\_\_ c. Enter From directory

1) Click on **Browse...** next to **From directory**

2) From the Import from directory window, select **<EMAILFILES>** and click **OK**

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

\_\_\_ d. Select the box next to **Customer.xsd and Order.xsd**

\_\_\_ e. Ensure that the **EmailFixedInboundModule** is selected for Into folder

\_\_\_ f. Click **Finish** from the Import window

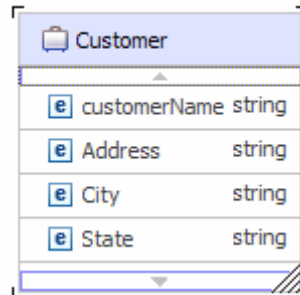
The Business Integration window is updated with the imported business objects.

\_\_\_ 3. Review the imported business objects:

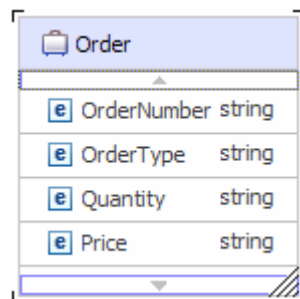
\_\_\_ a. Expand **EmailFixedInboundModule > Data Types** and you will now see two new data types '**Customer**' and '**Order**' under it.

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ b. Double-click on **Customer** review the fields inside the object:



\_\_\_ c. Now, double-click on **Order** and review the fields inside the object:



\_\_\_ 4. After reviewing, close the Customer and Order business objects from the Assembly editor

\_\_\_ 5. To start External Service from the Palette:

\_\_\_ a. From the **Palette** on the left side of Assembly Diagram, click on **Inbound Adapters**:

\_\_\_ b. Under Inbound Adapters, click on the **Email** and then click on the empty canvas of the assembly diagram. The New Flat File Service wizard is opened

---

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

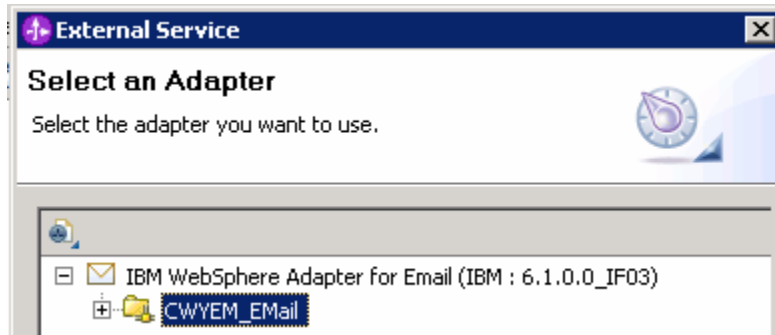
From the main menu, select **File > New > External Service**. This opens an External Service wizard that helps you obtain a service which establishes connectivity with other systems. The wizard provides three connectivity options – Adapters, Registers, and Messaging

Select the radio button next to **Adapters** and click **Next**

---

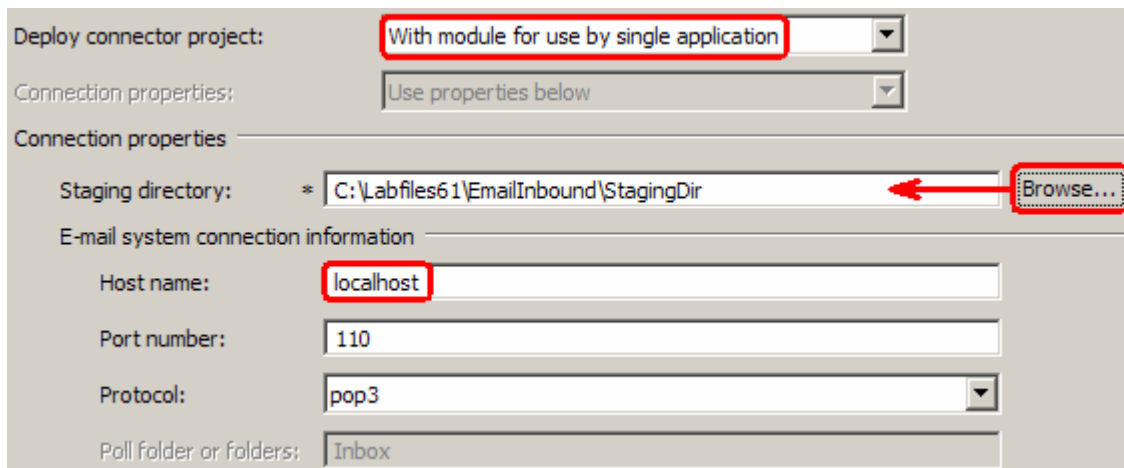
## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ 6. On the Select an Adapter screen, select **IBM WebSphere Adapter for Email (IBM : 6.1.0.0\_IF03)** and click **Next**



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

- \_\_\_ 7. Service Configuration Properties:
- \_\_\_ a. Deploy connector project: ensure that the default option **With module for use by single application** is selected
  - \_\_\_ b. Enter E-mail system connection information:
    - 1) Host name: **<EmailServer\_HostName>** (or IP Address of the machine that has E-mail Server), for Ex: localhost
    - 2) Port number: **110** (default, you should change it to the correct port number if your E-mail server is running on a different port)
    - 3) Protocol: accept the default **POP3**
    - 4) Not that the **Poll folder** is by default selected as **Inbox** and is not available
    - 5) For **Staging directory**, click on **Browse...** and select **<STAGING>** and click **OK**



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

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

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

\_\_ d. Event delivery configuration:

- 1) **Ensure once-only event delivery:** You should check this box only if you are using data source and table name in the Event persistence configuration (below). If this property is set to true, while using in-memory capability (explained below), the adapter will log a warning message. By default this is selected and you can accept the default selection.

\_\_ e. Event persistence configuration:

- 1) Ensure that the **Auto create event table** is **checked**
- 2) Event recovery table name: **EmailFixedTable**
- 3) Event recovery data source (JNDI) name: **jdbc/Email**

---

**Note:** This represents the JNDI name of the Data Source used by Event Persistence to get the JDBC database connection. The Data source must be created in the WebSphere Process Server. You should enter the data source JNDI name that you created in Step 3 of Part 1.

---

Event persistence configuration

Auto create event table

Event recovery table name:

Event recovery data source (JNDI) name:

User name used to connect to event data source:

Password used to connect to event data source:

Database schema name:

---

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

---

Advanced properties:

\_\_ f. User name: **username using which you connect to your E-mail server** (for Ex: Emailuser1@aimcp101.austin.ibm.com)

\_\_ g. Password: **password for the above user to connect to your E-mail server**

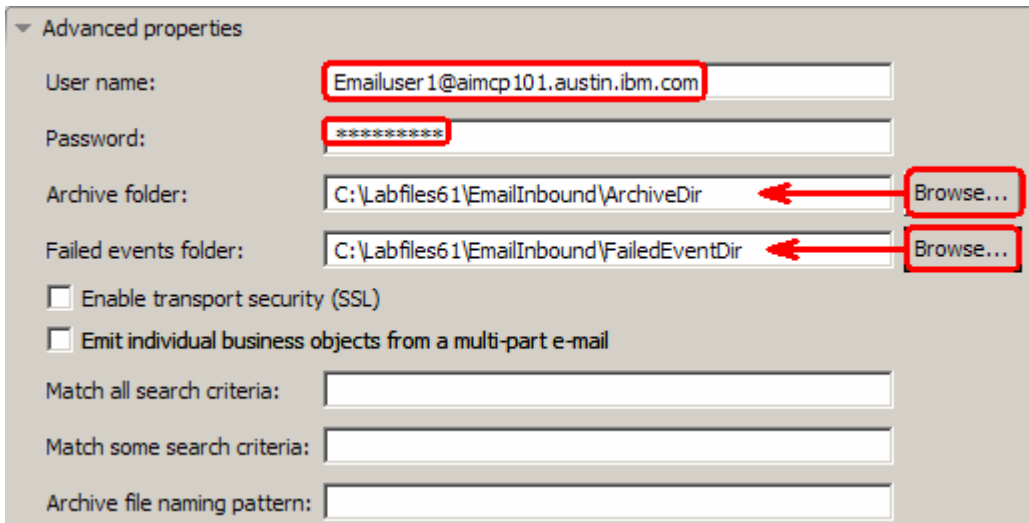
\_\_ h. Archive folder: All the processed events are archived in this folder

- 1) Click on **Browse...** next to it and select **<ARCHIVE>** and click **OK**

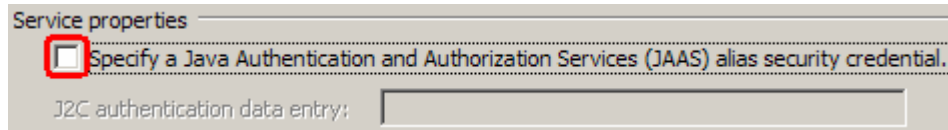
### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ i. Failed event folder: All the failed events are archived in this folder

1) Click on **Browse...** next to it and select **<FAILED>** and click **OK**



\_\_\_ 8. For this lab, you are not going to use the J2C authentication. So, uncheck the box next to Specify a Java Authentication and Authorization Services (JAAS) alias security credentials.



#### Function Selector Configuration:

\_\_\_ 9. Under Service properties, for **Function selector**, select **Use default function selector 'EmailFunctionSelector'** from the drop down list

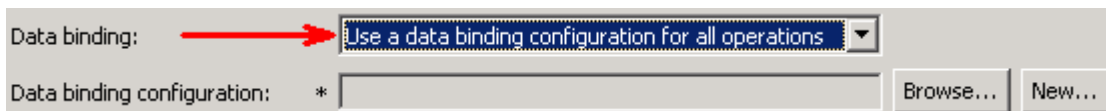
---

**Note:** Follow the steps in **Appendix: Define Function selector** to manually define a function selector.

---

\_\_\_ 10. You can define data binding in two places - service level (current screen of External Service wizard) or later at the method level (Operations screen of the External Service wizard). In this lab, you will define data binding at the service level (from this screen)

\_\_\_ a. From the dropdown menu next to Data binding, select **'Use a data binding configuration for all operations'**



#### Define Data binding configuration:

\_\_\_ b. Under **Specify the operation input**, click **New...** next to **Data binding configuration**. A Resource Configuration window is opened.



### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ c. Ensure that the selected module is **EmailFixedInboundModule**

1) For the **Name**, enter **EmailFixedDB**

The screenshot shows a configuration dialog box with the following fields and controls:

- Module:** A dropdown menu showing "EmailFixedInboundModule" and a "New..." button.
- Namespace:** A text field containing "http://EmailFixedInboundModule".
- Use default namespace**
- Folder:** An empty text field and a "Browse..." button.
- Name:** A text field containing "\*EmailFixedDB".

2) Click **Next**

\_\_ d. From the Select a Configuration Type screen, click on **Browse...** next to **Data binding class name**. A Data Binding Selection window is opened

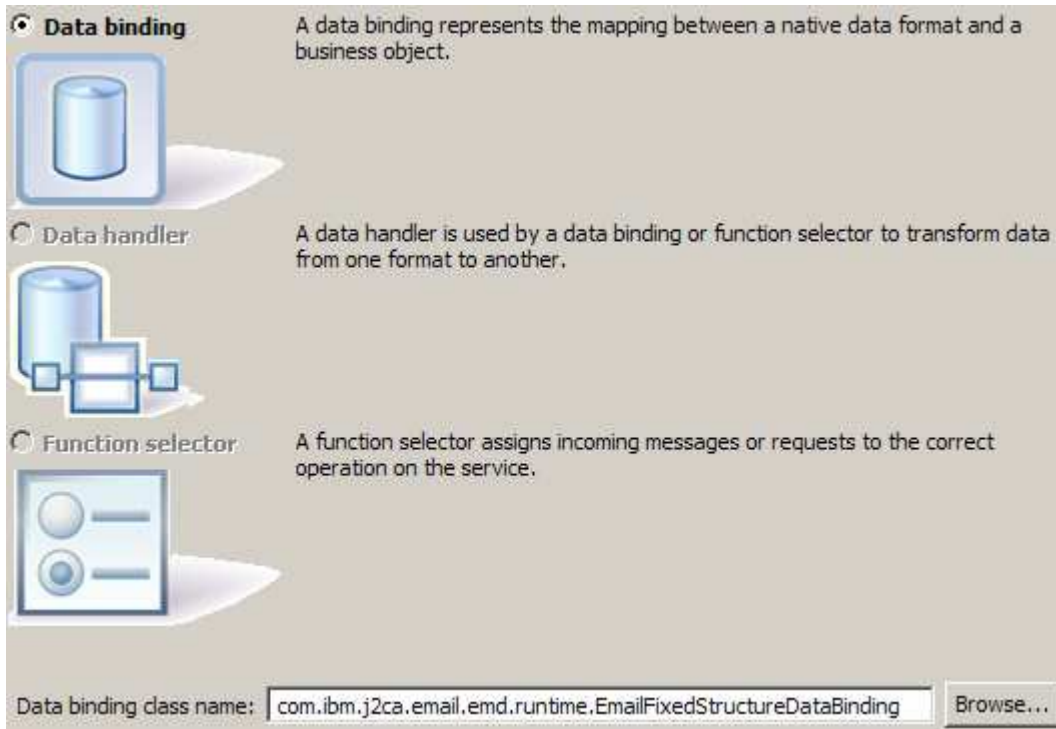
\_\_ e. Select **EmailFixedStructureDataBinding** under Matching data bindings and click **OK**

The screenshot shows the "Data Binding Selection" dialog box with the following elements:

- Show predefined data bindings:** Selected with a radio button.
- Show data binding classes:** Unselected with a radio button.
- Filter by name (? = any character, \* = any String):** A text field containing "\*".
- Matching data bindings:** A list box containing three items: "Email Simple DataBinding", "EmailFixedStructureDataBinding" (highlighted), and "EmailWrapperDataBinding".
- Qualifier:** A text field containing "com.ibm.j2ca.email.emd.runtime.EmailFixedStructureDataBinding".
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ f. Above selected data binding,  
**com.ibm.j2ca.email.emd.runtime.EmailFixedStructureDataBinding**, is populated:



- \_\_\_ g. Click **Next**

- \_\_\_ 11. From **Data Binding Properties** screen,

- \_\_\_ a. Define **mailContent**:

- 1) Click **Add...** next to the table. Add/Edit window is opened
- 2) Note that the **E-mail part** is **mailContent**
- 3) Click on **Browse...** next to **Business object type**. Data Type Selection window is opened
- 4) Select **Customer** from the **Matching data types** list and click **OK**

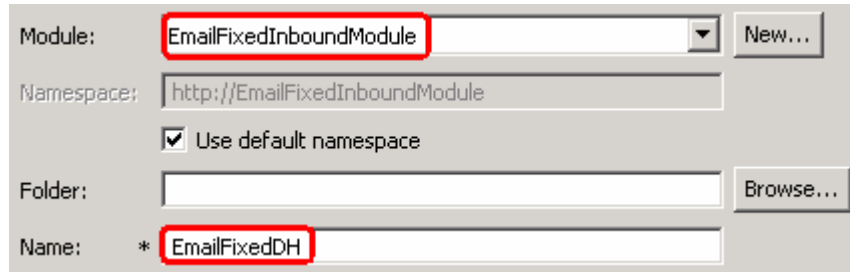
- \_\_\_ b. For **Mime type**, select **text/xml** from the drop down list

### Data Handler Configuration:

- \_\_\_ c. Click **New...** next to **Configured data handler**. A new Resource Configuration window is opened for you to define the data handler

### IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_ d. Ensure that the module selected is **EmailFixedInboundModule** and enter **EmailFixedDH** for the **Name** of the data handler that is created



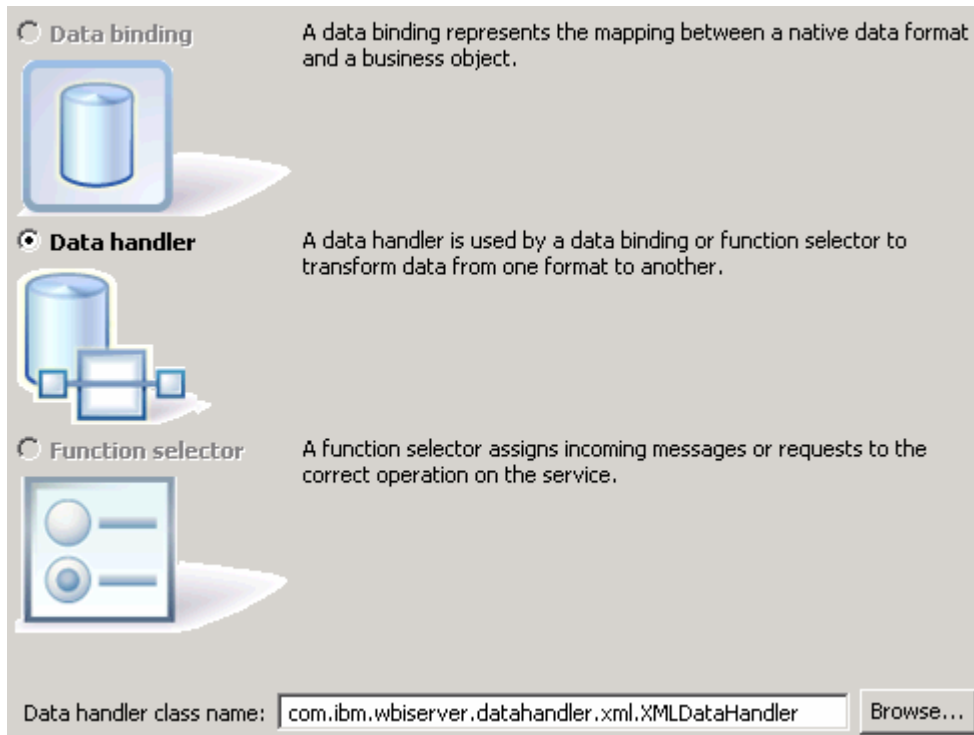
- \_\_ e. Click **Next**

By default, the radio button next to **Data Handler** is selected

- \_\_ f. Click on **Browse...** next to **Data handler class name**. Data Handler Selection window is opened.

- \_\_ g. Select **XML Data Handler** from the Data Handlers list and click **OK**

- \_\_ h. You are now back to Binding Resource Configuration window and the above selected data handler, **com.ibm.wbiserver.datahandler.xml.XMLDataHandler**, is displayed as the Class Name of data handler:

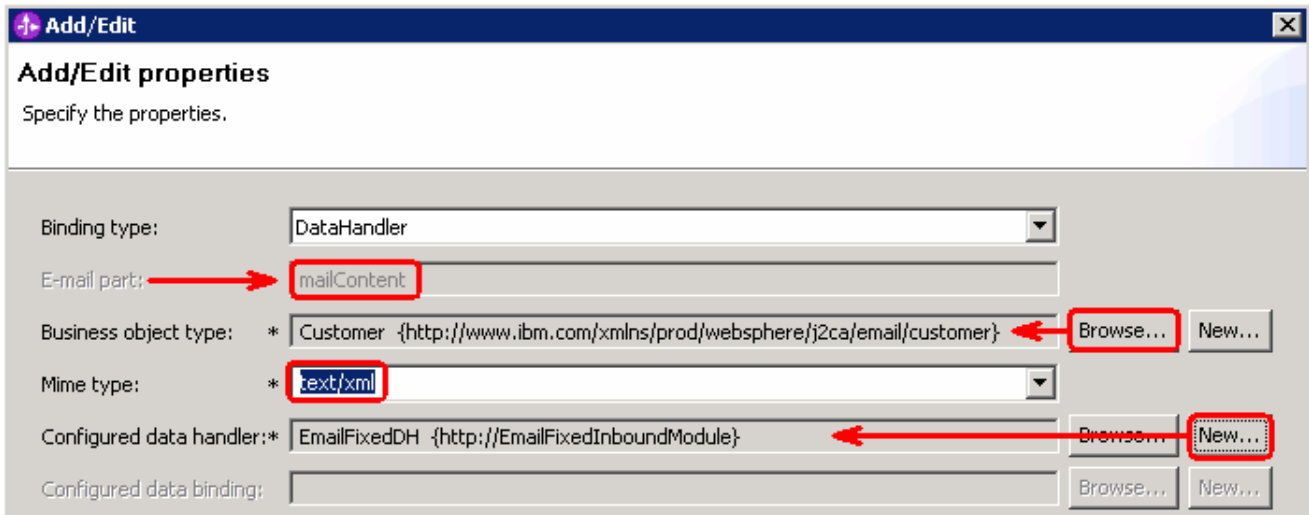


- \_\_ i. Click **Next**

- \_\_ j. Accept the default selection '**UTF-8**' for encoding and click **Finish**

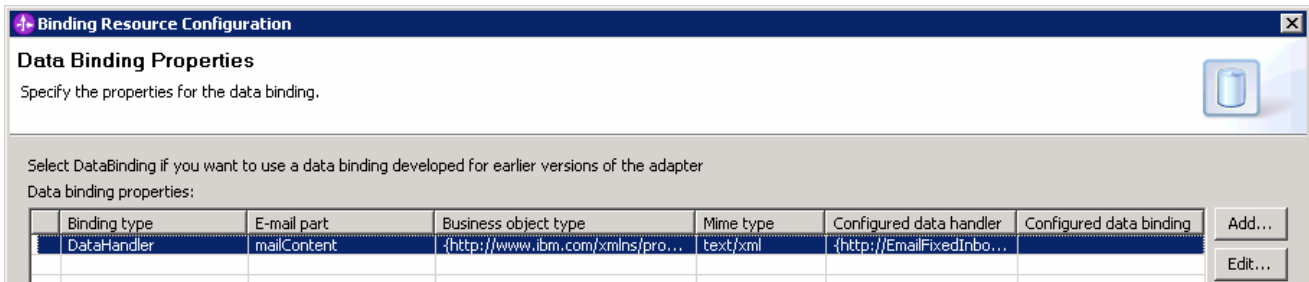
IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_ k. You are now done with defining the data handler and back to Add/Edit properties screen. The Data handler configuration name, **EmailFixedDH** is populated in this screen



- \_\_ l. Click **Finish** from the Add/Edit properties screen

- \_\_ m. Now, back to Binding Resource Configuration window. You have so far defined the Business object type for the mailContent which is populated in this window:



- \_\_ n. Define **attachment** part:

- 1) Click on **Add...** next to the 'Data binding properties' table. This will open an Add/Edit window
- 2) Note that the **E-mail part** is **attachment1**
- 3) Click on **Browse...** next to **Business object type**. Data Type Selection window is opened

- \_\_ o. Select **Order** from the Matching data types list and click **OK**

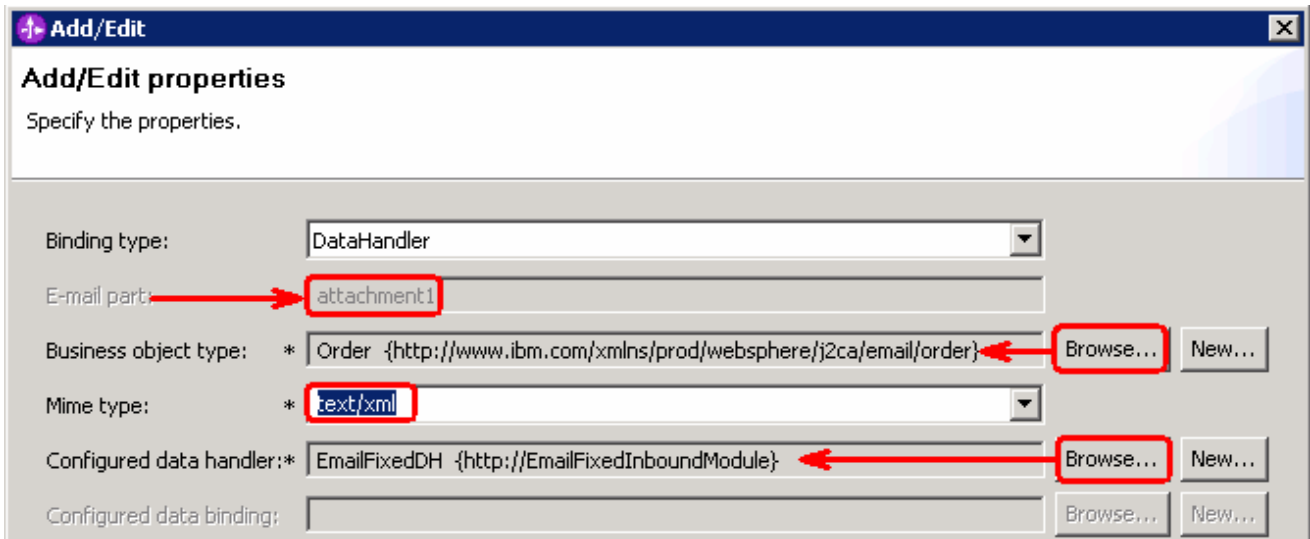
- \_\_ p. For **Mime type**, select **text/xml** from the drop down list

- \_\_ q. Click **Browse...** next to **Configured data handler**

- 1) From the Data Handler Selection window, select **EmailFixedDH** and click **OK**

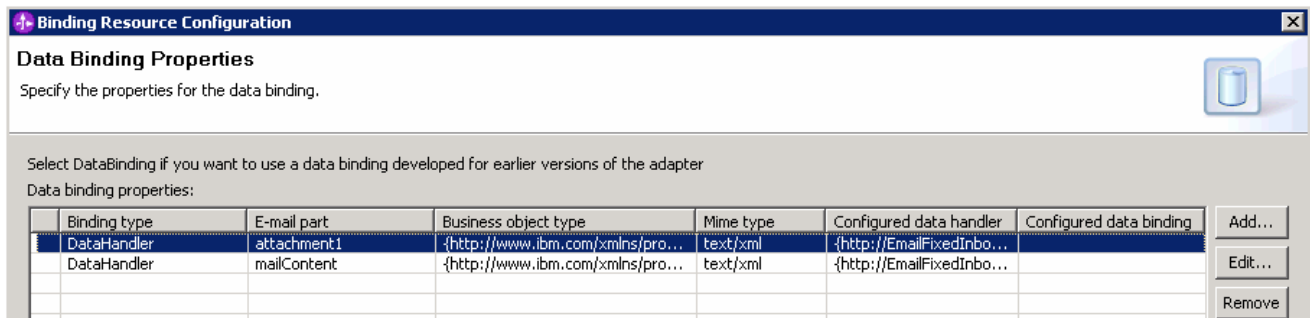
IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ r. The data handler is populated in Add/Edit window:



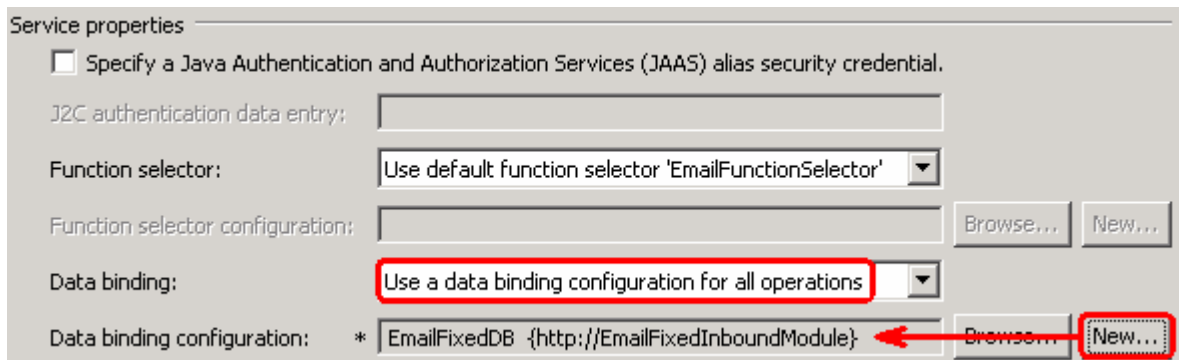
\_\_ s. Click **Finish** from the Add/Edit window

\_\_ t. You are back to the Binding Resource Configuration window and the defined Business object type and the Email-part are displayed in the table format in this screen:



\_\_ u. Click **Finish** from the Data Binding Properties screen

\_\_ v. Now the **EmailFixedDB** is displayed for Data binding configuration



\_\_\_ 12. Check the box next to **Change logging properties for wizard** to view the output location of the log file and the logging level. You can change the logging level using the drop down menu

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ a. Click **Next**

Following screen is the Operations screen where you can define all your operations.

In V6.1, you can select from three different Data types for any operation:

- Generic e-mail business object
- Generic e-mail business object with business graph
- User defined type

In this part of the lab, you will use the last option, **User defined type**.

### Create emitFixedEmail Operation:

\_\_\_ 13. Click on **Add...** to open Add Operation window

\_\_\_ a. For **Data type for the operation input**, select **User defined type** from the drop down list



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

The Data type for input is populated based on the selection of the Data type for the operation in the previous step. Since you have chosen User defined type, the Data type for input is **blank**. You can define your own data type here.

\_\_\_ c. For Operation name, enter **emitFixedEmail**

#### Define Input type:

\_\_\_ d. Click on **New...** next to **Input type**. New Business Object is opened

\_\_\_ e. Ensure that the selected Module is **EmailFixedInboundModule** and click **Next**

\_\_\_ f. For **Business object name**, enter **MyCustomEmail** (you can enter any name here)

\_\_\_ g. **Select** the box next to **Generate a business graph for the business object**

#### Define mailContent:

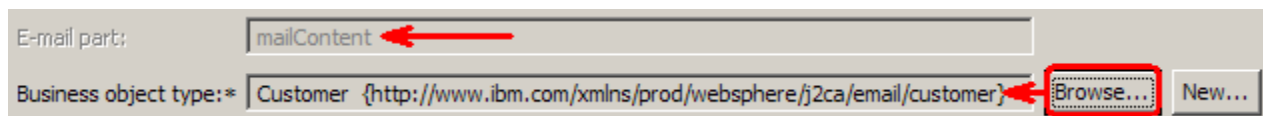
\_\_\_ h. Click on **Add...** next to the 'Properties for e-mail type' table. This will open an Add/Edit window

1) Note that the **E-mail part** is **mailContent**

2) Click on **Browse...** next to **Business object type**. Data Type Selection window is opened

3) Select **Customer** from the Matching data types list and click **OK**

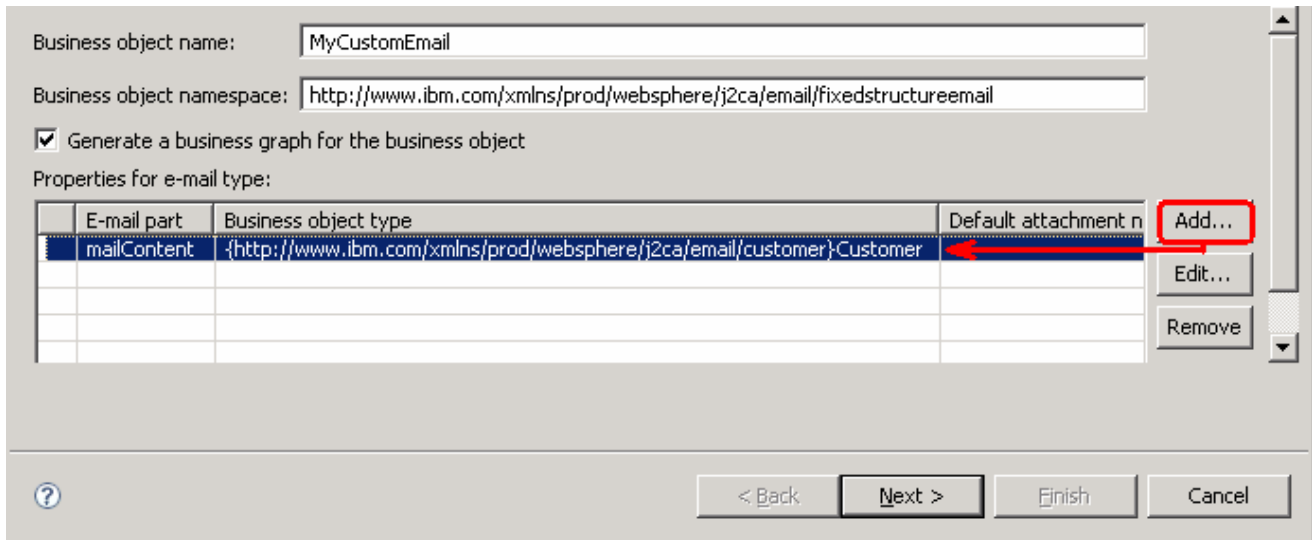
\_\_\_ i. Now, back to Add/Edit window. You have so far defined the Business object type for the mailContent which is populated in this window:



IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_ j. Click **Finish** from the Add/Edit window

\_\_ k. You are back to the New Business Object window and the defined Business object type and the Email-part are displayed in the table format in this screen:

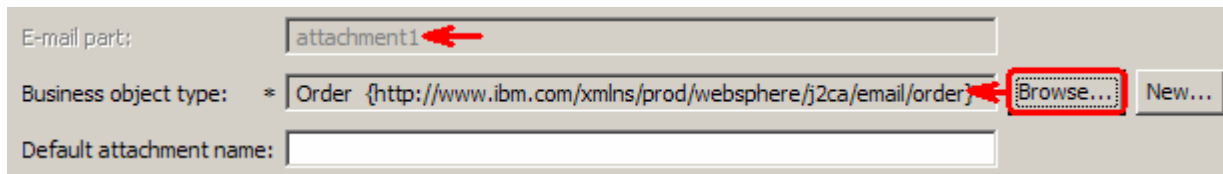


Define **attachment** part:

\_\_ l. Click on **Add...** next to the 'Properties for e-mail type' table. This will open an Add/Edit window

- 1) Note that the **E-mail part** is **attachment1**
- 2) Click on **Browse...** next to Business object type. Data Type Selection window is opened
- 3) Select **Order** from the Matching data types list and click **OK**

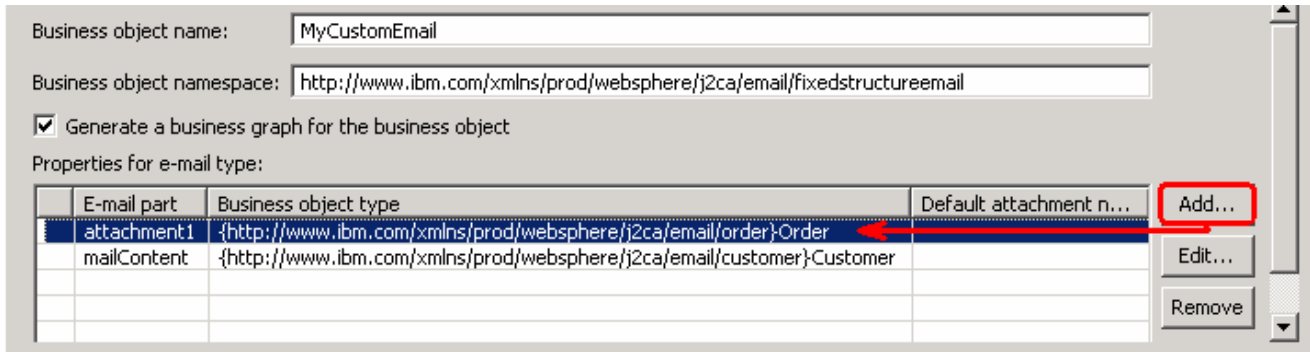
\_\_ m. Now, back to Add/Edit window. You have so far defined the Business object type for the attachment1 which is populated in this window:



\_\_ n. Click **Finish** from the Add/Edit window

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

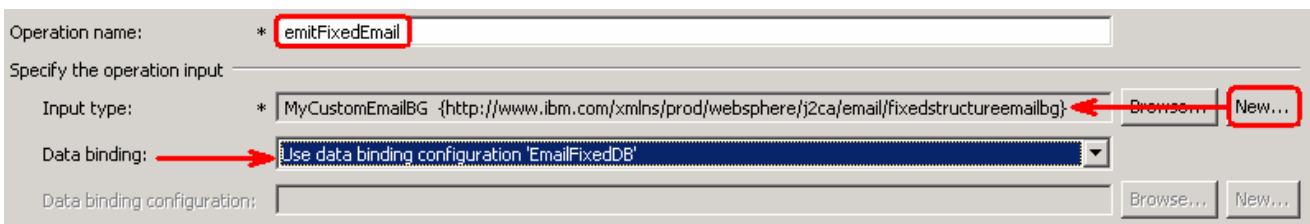
\_\_ o. You are back to New Business Object window and the defined Business object type and the Email-part are displayed in the table format in this screen:



\_\_ p. Click **Finish**

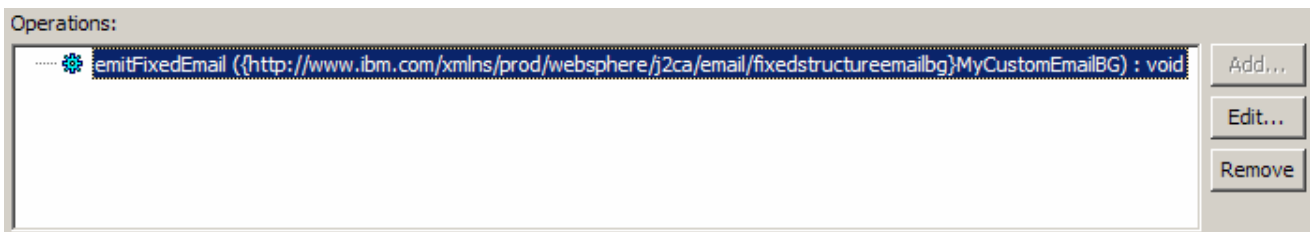
You are now back to the Add operation window and the Input type you have defined in the previous steps is displayed here.

\_\_ q. For Data binding, accept the default selection **Use data binding configuration 'EmailFixedDB'**



\_\_ r. Click **Finish** from the Operation screen

\_\_\_ 14. The above defined operation, **emitFixedEmail**, is populated under Operations list



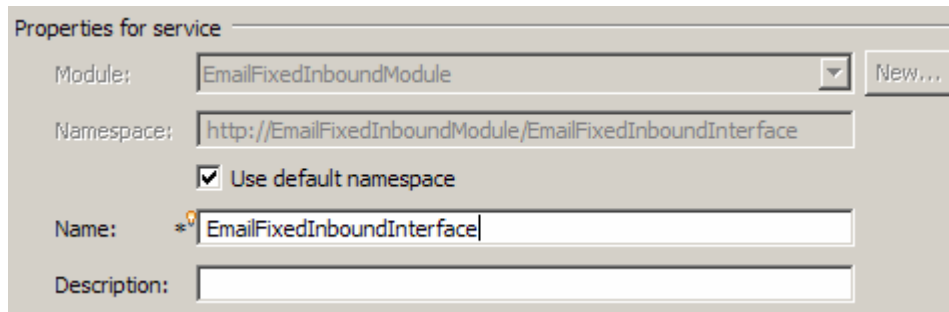
\_\_ a. Click **Next** from Operations screen



## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ 15. From Generate Service screen:

\_\_\_ a. For Name, enter **EmailFixedInboundInterface**



Properties for service

Module:

Namespace:

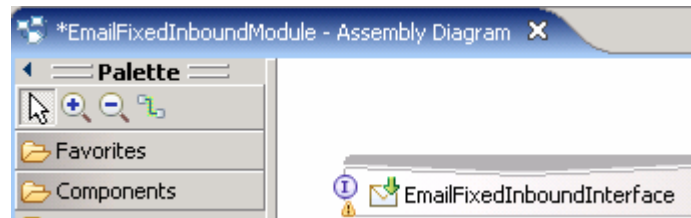
Use default namespace

Name:

Description:

\_\_\_ b. Click **Finish**

\_\_\_ 16. The Assembly diagram for EmailFixedInboundModule is opened with an export component, EmailFixedInboundInterface:





\_\_\_ a. Save (**Ctrl + S**) changes to your assembly diagram

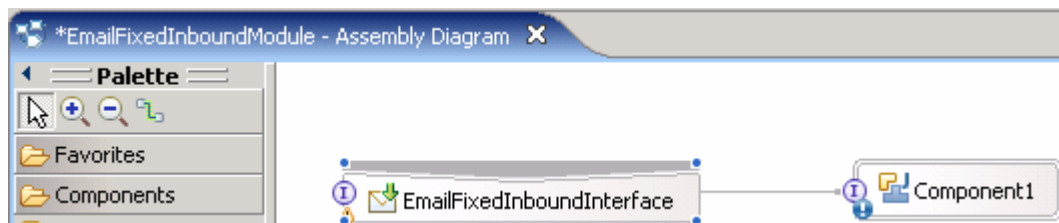
## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

## 4.2. Add Java component

In this part of the lab, you will add a Java component and then wire the component to the existing export interface. Then you will continue to add a simple print Java code to the added Java component.

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

Your assembly diagram for EmailFixedInboundModule will look like this:



- \_\_\_ 4. Generate implementation for Component1
  - \_\_\_ a. Right-click on **Component1** and select **Generate Implementation** from the pop-up menu
  - \_\_\_ b. On the **Generate Implementation** panel, select **default package**, and click **OK**
  - \_\_\_ c. **Component1Impl.java** is opened in Assembly editor. Scroll down to the method **emitEmailPS(DataObject emitEmailPSInput)** that needs to be implemented and add this code under that method:

```
public void emitEmailPS(DataObject emitEmailPSInput) {
    //TODO Needs to be implemented.
}
```

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

```

System.out.println("*****ENDPOINT
emitFixedEmail*****");

DataObject customemail =
emitFixedEmailInput.getDataObject("MyCustomEmail");

System.out.println("*****Email content*****");

DataObject customer = customemail.getDataObject("mailContent");

String name = customer.getString("CustomerName");

System.out.println("NAME-----> "+name);

String address = customer.getString("Address");

System.out.println("ADDRESS--> "+address);

String city = customer.getString("City");

System.out.println("CITY-----> "+city);

String state = customer.getString("State");

System.out.println("STATE-----> "+state);

System.out.println("*****Attachment data*****");

DataObject order = customemail.getDataObject("attachment1");

String OrderNumber = order.getString("OrderNumber");

System.out.println("ORDER NUMBER-----> "+OrderNumber);

String OrderType = order.getString("OrderType");

System.out.println("ORDER TYPE--> "+OrderType);

String Quantity = order.getString("Quantity");

System.out.println("QUANTITY-----> "+Quantity);

String Price = order.getString("Price");

System.out.println("PRICE-----> "+Price);

```

---

**Note:** The Java code is also available at <EMAILFILES>\ FixedInbondJavaCode.txt

---

- \_\_ d. Save (**Ctrl + S**) and close Component1Impl.java
- \_\_ e. Save (**Ctrl + S**) and close Assembly diagram: EmailFixedInboundModule

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

---

### 4.3. Test user defined scenario

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

- \_\_\_ 1. Start your e-mail server (if not started already)
  - \_\_\_ a. Select **Start > hMailServer >hMailServerAdministrator**
  - \_\_\_ b. From the hMailServer Administrator – Connect window, ensure that **localhost** is selected and click on **Connect**
  - \_\_\_ c. hMailServer Administrator window is opened and the Current status should show **Running**
- \_\_\_ 2. Add the project to the WebSphere Test Environment server
  - \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
  - \_\_\_ b. From the Add and Remove Projects window, select **EmailFixedInboundModuleApp** under Available projects panel and click **Add >**
  - \_\_\_ c. You will now see the **EmailFixedInboundModuleApp** added to the **Configured projects**
  - \_\_\_ d. Click **Finish** and wait until the project is being published onto the server. The server will start in Debug mode if it is not already started before
- \_\_\_ 3. Review and modify the batch file to generate required event E-mails
  - \_\_\_ a. From windows explorer, browse to **<EMAILFILES>EmailEventCreator**

---

**Note:** You can find EmailEventsCreator.zip at <EMAILRAR\_FILE>\Samples also which shipped with WebSphere Integration Developer. You will need to extract it and make modifications to the batch file inside the archive file. For your convenience, you can use the files under **<EMAILFILES>EmailEventCreator** which are already modified for this lab.

---

- \_\_\_ b. Review the batch file, createEvents.bat, under EmailEventsCreator folder:

Structure of **createEvents.bat** file:

```
<host> <port> <number of events to be created> <from> <to> <cc> <bcc> <subject> <content>
<path for event file name> <Boolean for pass through/non pass through>
```

Where, Path for event file name - is the path on the local folder where the event file attachment has been placed.

Boolean for pass-through/non-pass-through – has true or false values. **True** indicates pass-through and **False** indicates non-pass-through.

- \_\_\_ c. Review **createEvents\_sample\_fixed.bat** file under EmailEventsCreator folder. This file, for your convenience, is created by providing the required inputs specified in the createEvents.bat. This is used to generate the required E-mail events:

```
java -cp .\;\mail.jar;\activation.jar EmailEventCreatorNew "localhost" 25 1
"Emailuser2@aimcp101.austin.ibm.com" Emailuser1@aimcp101.austin.ibm.com
Emailuser3@aimcp101.austin.ibm.com Emailuser4@aimcp101.austin.ibm.com "Test Email"
```

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

```
inbound Fixed" "C:\Labfiles61\EmailFiles\Customer.xml" "C:\Labfiles61\EmailFiles\Order.xml"
>false"
```

---

**Note:** This will generate 1 event e-mail from [Emailuser2@aimcp101.austin.ibm.com](mailto:Emailuser2@aimcp101.austin.ibm.com) on the localhost mail server, to the inbox of [Emailuser1@aimcp101.austin.ibm.com](mailto:Emailuser1@aimcp101.austin.ibm.com) with the file **Order.xml** as an attachment and the content of **Customer.xml** as the body of the e-mail. **You should change the fields according to your e-mail server settings and file locations.**

---

- \_\_\_ 4. Run the batch file to generate E-mail
- \_\_\_ a. Open a command prompt and change the directory to **<EMAILFILES>EmailEventCreator**
  - \_\_\_ b. Enter **createEvents\_sample\_fixed.bat**
  - \_\_\_ c. You can see the E-mail in the inbox of Emailuser1, but it is deleted quickly by the adapter after polling.
- \_\_\_ 5. Verify the results:
- \_\_\_ a. Because the adapter is running, the event E-mail will pass through the **emitFixedEmail** method and you should see these messages in the console (or SystemOut.log):

```
[10/16/07 15:34:26:718 CDT] 0000023c logging      I   Start invoking
EventPointImpl.getEventSource

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O
*****ENDPOINT emitFixedEmail*****

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O *****Email
content*****

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O NAME-----> IBM

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O ADDRESS--> 11501
Burnet Rd

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O CITY-----> Austin

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O STATE-----> TX

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O
*****Attachment data*****

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O ORDER NUMBER----->
ABC12345

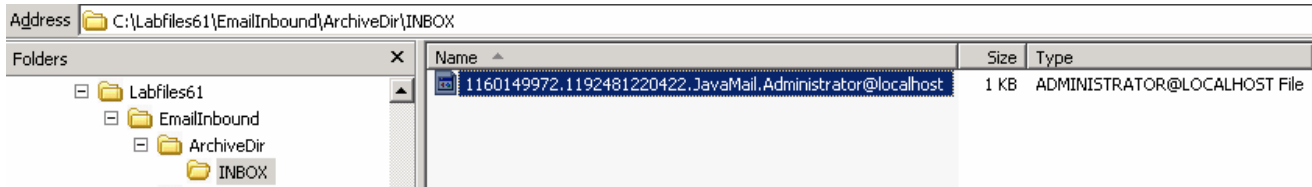
[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O ORDER TYPE--> BULK

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O QUANTITY-----> 500

[10/16/07 15:34:26:984 CDT] 00000061 SystemOut    O PRICE-----> 26.59
```

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ b. Check the **ArchiveDir** which should contain a folder **INBOX** (created by the adapter) and if you open that **INBOX** folder, you will see a new file with the name ending in your host name



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

- \_\_\_ a. Right-click on **WebSphere Process Server V6.1** under the Servers view and select **Add and remove projects...** from the pop-up menu
- \_\_\_ b. Select **EmailFixedInboundModuleApp** 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

## IBM WEBSPHERE ADAPTER 6.1 – LAB EXERCISE

### **What you did in this exercise**

In this lab, you started with importing the E-mail Adapter RAR file into your WebSphere Integration Developer new workspace. Then, used your WebSphere Process Server administrative console and configured it to create the Data source and data base required to complete this lab.

You made use of the External 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 pass-through test scenarios and two content specific (non pass through) test scenarios.

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

## Appendix: Define function selector

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

- \_\_\_ 1. Under Service properties, for **Function selector**, select **Use default function selector configuration** from the drop down list
- \_\_\_ 2. Click on **New** next to **Function selector configuration**. The Resource Configuration window is opened.
  - \_\_\_ a. Ensure that the selected module is **EmailPSInboundModule**
  - \_\_\_ b. For Name, enter any string. For Ex: **PSFS**

Module: EmailPSInboundModule [New...]

Namespace: http://EmailPSInboundModule  Use default namespace

Folder: [Browse...]

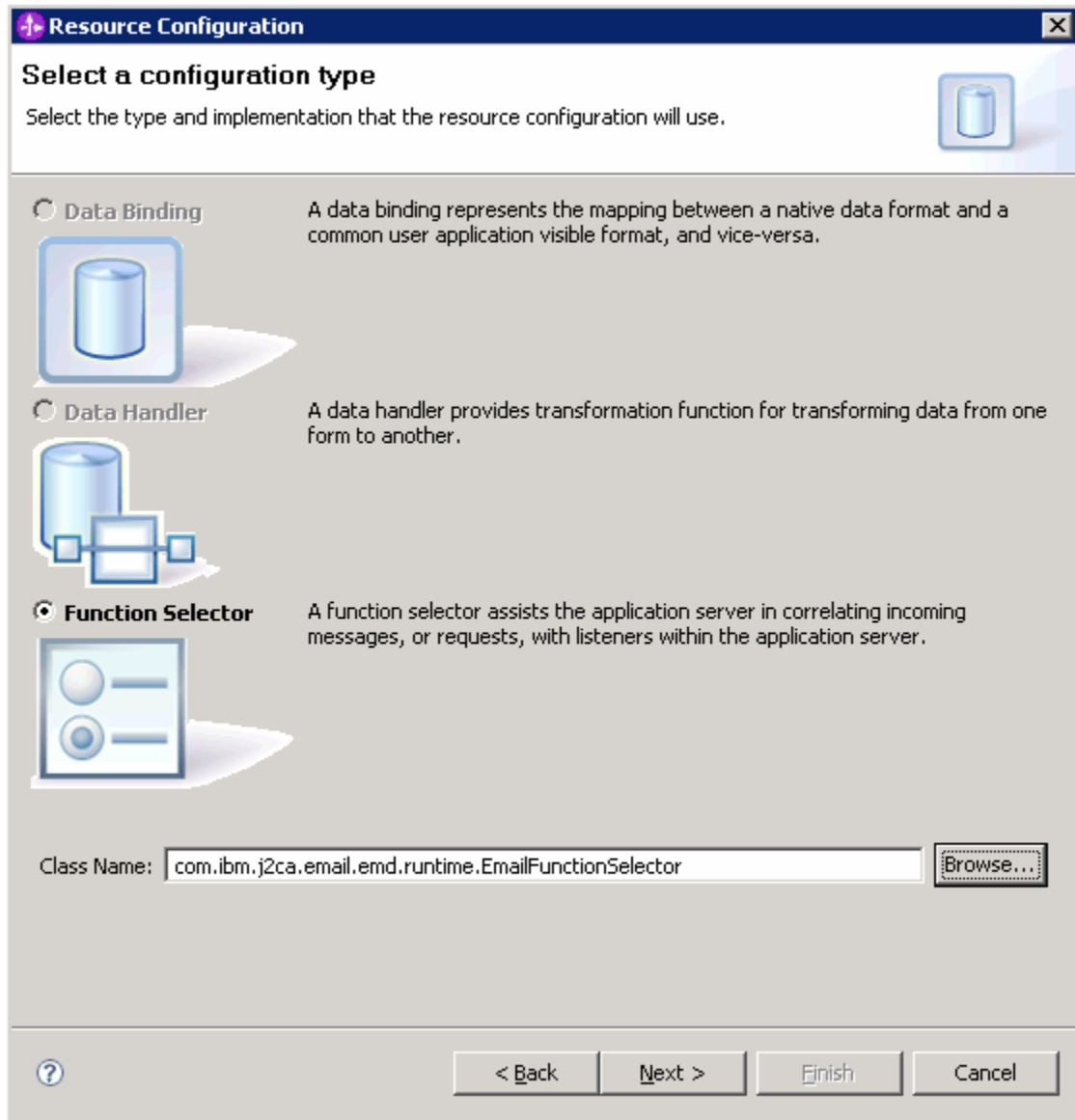
Name: \* PSFS

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



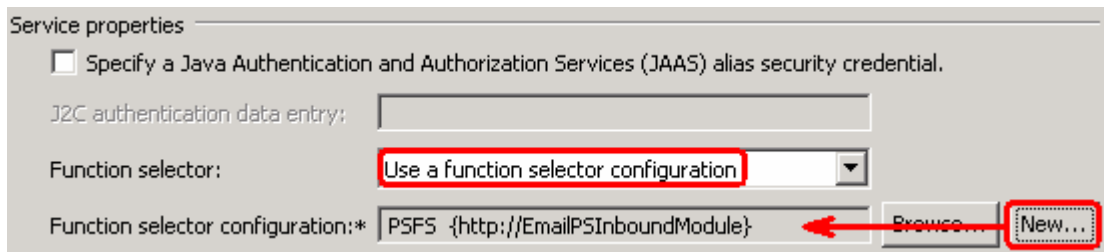
IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ 3. In the next 'Select a configuration type' screen, accept the default Function Selector Class Name, **com.ibm.j2ca.extension.emd.runtime.EmailFunctionSelector** :



- \_\_\_ a. Click **Finish**

- \_\_\_ 4. You will now be back to External Service window and the function selector created in the above steps is populated:



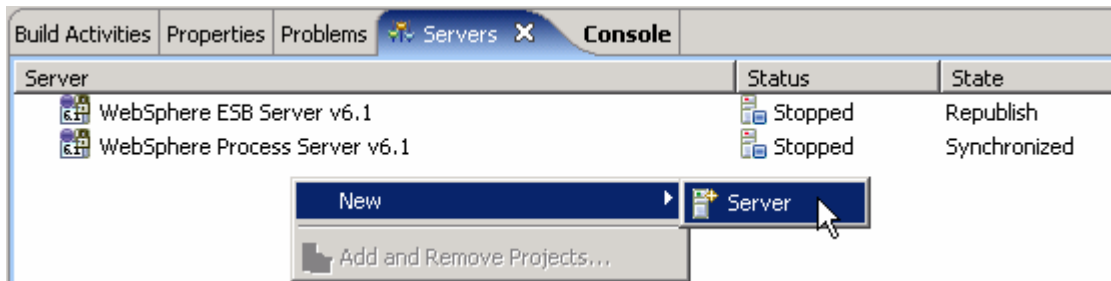
## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

## Task: Adding remote server to WebSphere Integration Developer test environment

This task describes how to add a remote server to the WebSphere Integration Developer Test environment. This example uses a z/OS machine.

Create a new remote server.

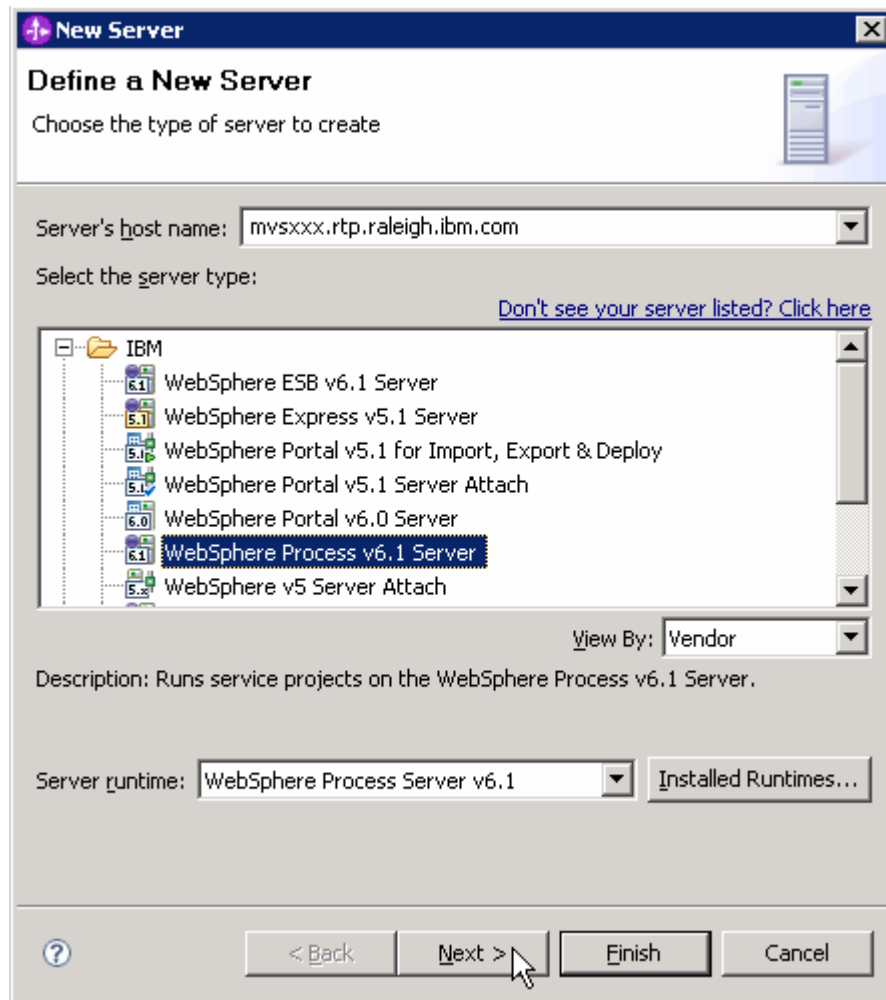
- \_\_\_ 1. Right click on the background of the Servers view to access the pop-up menu.
  - \_\_\_ a. Select **New > Server**.



- \_\_\_ 2. Enter these from New Server window:
  - \_\_\_ a. In the New Server dialog, specify the remote server's host name, **<HOSTNAME>**

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ b. Ensure that the appropriate server type, '**WebSphere Process v6.1 Server**' or '**WebSphere ESB v6.1 Server**', is highlighted in the server type list



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

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

- \_\_\_ 3. On the WebSphere Server Settings page, leave the radio button for **SOAP** selected, changing the **SOAP connector port** to the correct setting (<SOAP\_PORT>). If security is on in your server, check the box for 'Security is enabled on this server' and input <USERID> for the user ID and <PASSWORD> for the password.

**New Server**

**WebSphere Server Settings**  
Input settings for the new WebSphere server.

WebSphere profile name: \_\_\_\_\_

Server connection type and admin port

RMI (Designed to improve communication with the server)  
ORB bootstrap port: 2809

SOAP (Designed to be more firewall compatible)  
SOAP connector port: 8880

Run server with resources within the workspace

Security is enabled on this server

Current active authentication settings:

User ID: ssadmin

Password: ●●●●●●

Server name: sssr011

Server type

BASE, Express or unmanaged Network Deployment server

Network Deployment server

Network Deployment server name: \_\_\_\_\_

The server name is in the form of:  
<cell name>/<node name>/<server name>  
For example, localhost/localhost/server1. In a cluster environment,  
the server name is in the form of:  
<cell name>/<cluster name>

Click this button to detect the server type.

- \_\_\_ a. Click **Finish**.

## IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

\_\_\_ 4. The new server should be seen in the Server view

Server	Status	State
WebSphere ESB Server v6.1	Stopped	Republish
WebSphere Process Server v6.1	Stopped	Synchronized
WebSphere Process v6.1 Server @ mvsxxx.rtp.raleigh.ibm.com	Stopped	Republish

\_\_\_ 5. Start the remote server if it is not already started.

---

**Note:** WebSphere Integration Developer V6.1 does not support starting remote servers from the Server View. So, you should start the remote server from the remote machine where the server is installed.

---

\_\_\_ a. From a command prompt, telnet to the remote system if needed (for z/OS or i5/OS):

**'telnet <HOSTNAME> <TELNET\_PORT>'**

user ID : <USERID>

password : <PASSWORD>

\_\_\_ b. Navigate to the bin directory for the profile being used:

**cd <WAS\_HOME>/profiles/<PROFILE\_NAME>/bin**

\_\_\_ c. Run the command file to start the server: **./startServer.sh <SERVER\_NAME>**

\_\_\_ d. Wait for status message indicating server has started:

```
ADMU3200I: Server launched. Waiting for initialization status
ADMU3000I: Server cllsr01 open for e-business; process id is 00000120000000002
```

IBM WEBSHERE ADAPTER 6.1 – LAB EXERCISE

This page is left intentionally blank.