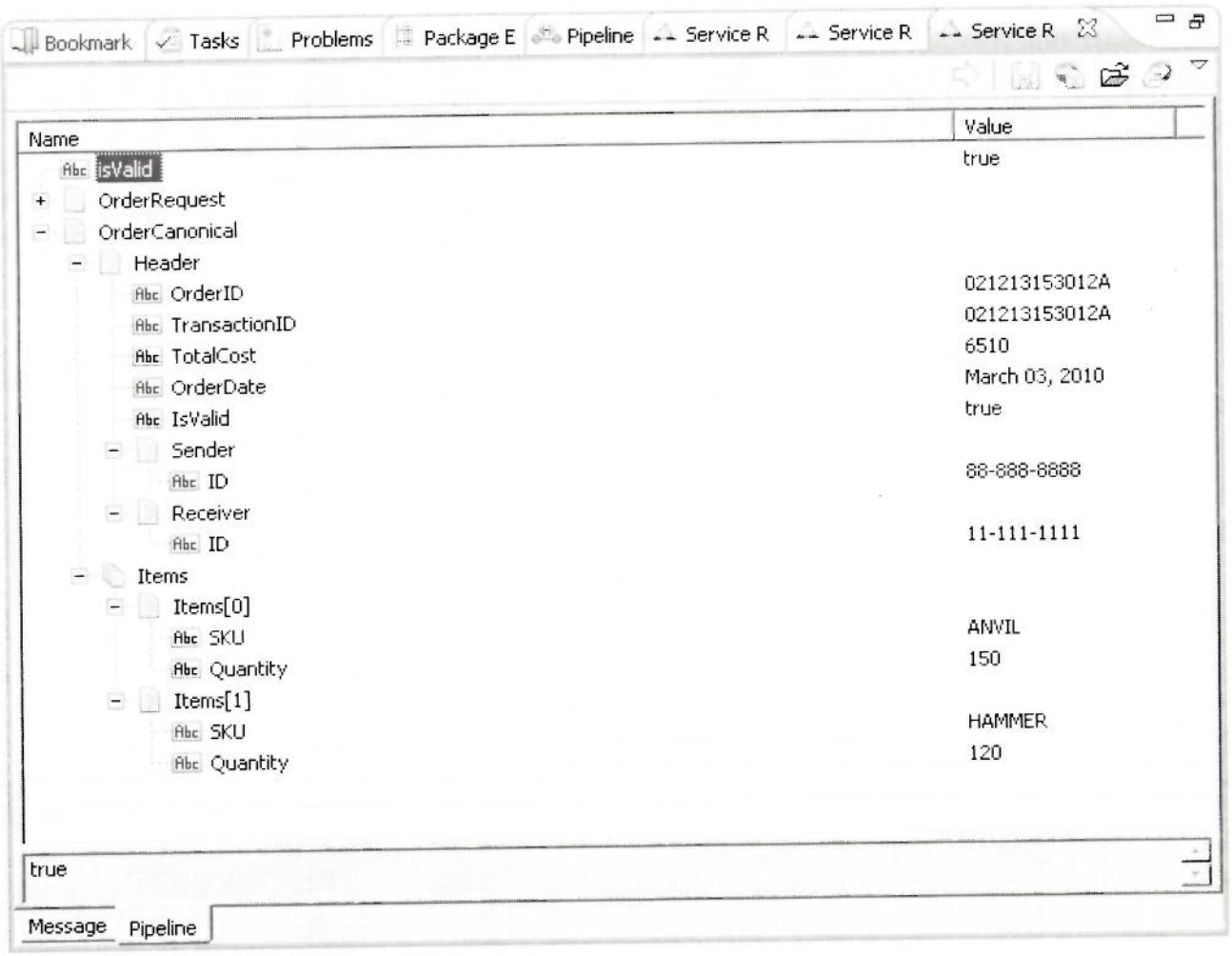
8. Save the service and run. Use the Load button and the input file **...\IntegrationServer\packages\AcmeSupport\pub\order_request_input.txt** (Do not forget to set **isValid** to true or false when you test!).

   Check the **Results** panel. Collapse **OrderRequest** and look at **OrderCanonical.** This variable must be completely populated. Especially check the date and the uppercase **OrderID**, **TransactionID**, and **SKU** values.



Check Your Understanding

1. How is a transformer different from a normal service?
2. What if the transformer you want to use is not in the transformer drop-down list?
3. Why did we need to LOOP over ProductLineItems? Why not just map from ProductLineItems to Items?

This page intentionally left blank

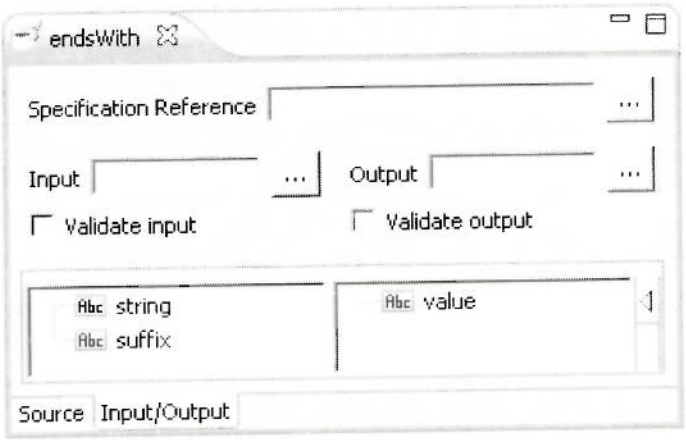
Exercise 10:
Create a Java Service

Overview

In this exercise, you will create, compile, and run a Java Service using Designer. Imagine that you require a special service that tests if a string ends in a second string. There is no such service in the pub.string folder, but you want to use the String.endsWith() method in the Java runtime environment.

Steps

1. Create a new Java service called **endsWith** in the **acme.PurchaseOrder.work** folder. This service has two string inputs, called **string** and **suffix**.



2. Enter the following code for your service:

```
IDataCursor cursor = pipeline.getCursor();

String string = IDataUtil.getString(cursor, "string");
String suffix = IDataUtil.getString(cursor, "suffix");

String value = string.endsWith(suffix) ? "true" : "false";

IDataUtil.put(cursor, "value", value);

cursor.destroy();
```

   Note: All Java development features, like code completion, that you are used to from the Eclipse IDE are available.

3. Run your service with some sample input values and verify that the returned values are correct.

Check Your Understanding

1. What exactly is each line of the Java code doing in the endsWith service?
2. Is the service thread safe? What would you have to do if not?
3. How could the cursor handling be improved?

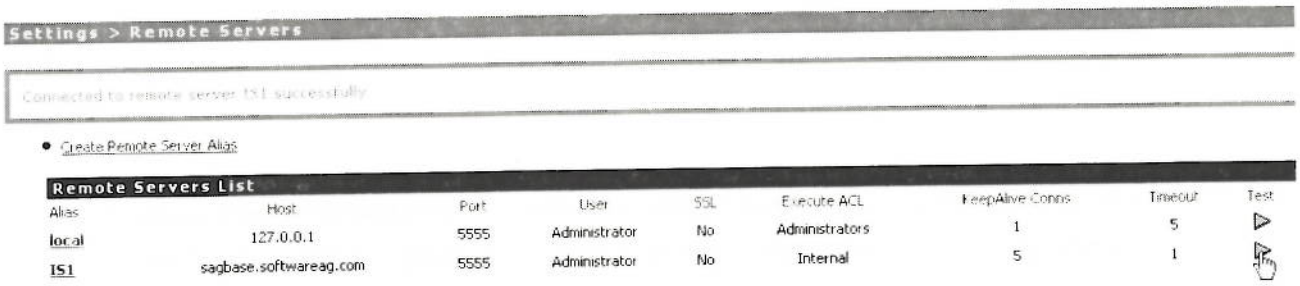
Exercise 11:
Monitoring Services

Overview

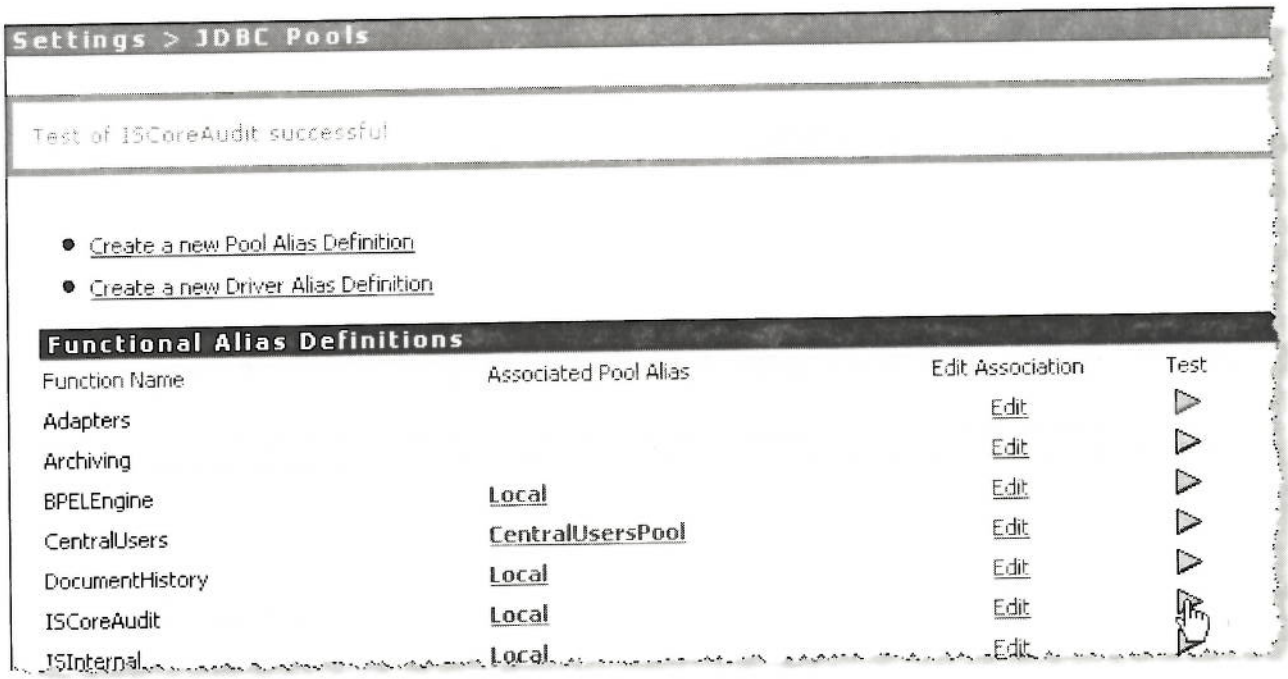
In this Exercise, you will use My webMethods to track the execution of services.

Steps

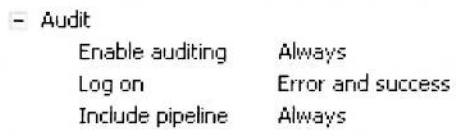
1. Open the IS Administrator console (*http://localhost:5555*) and log in as "Administrator" using a password of "manage". In the **"Settings"** menu select **Remote Servers** and verify that there is an entry for IS1 in the **Remote Servers List**. Test the IS1 alias by clicking on its Test icon (▷).



2. Go to the Settings ➡ JDBC Pools entry area in the Administrator console and confirm that a **JDBC Pool Alias** named **Local** is **Associated** with **ISCoreAudit, ISInternal, ProcessAudit** and **ProcessEngine** Functional Aliases. Click on the **Test** button (▷) to the right of the **ISCoreAudit** Functional Alias in order to confirm your connection to the database is working.



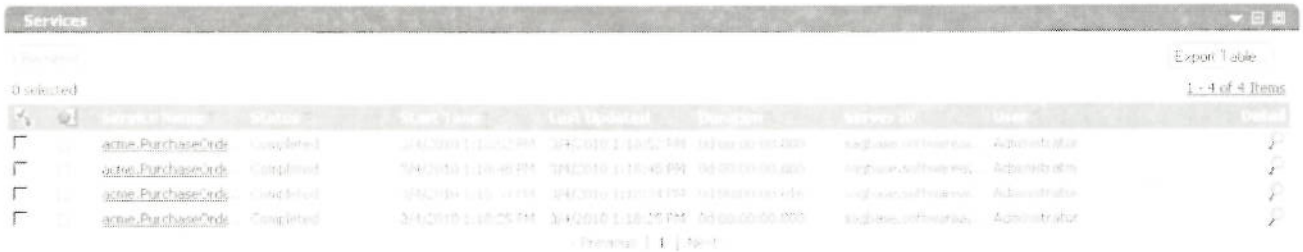
3. If you changed any values in the previous 2 steps, then restart the Integration Server. If you did not change any values, continue without restarting.
4. In Designer, open the service **acme.PurchaseOrder.work:branch1**. In the Service Properties, enable Audit Logging. Set the Audit properties as follows:
   a. Enable Auditing = Always
   b. Log on = Error and Success
   c. Include Pipeline = Always



5. Save the **branch1** service and run it several times. Provide different input strings during each run.
6. Open the My webMethods console (http://localhost:8585) in a browser and log in as **Administrator | manage**.
7. Note: When using the My webMethods console for the first time on a freshly started MWS, the response times can be inappropriately long. This is caused by the fact that MWS has to load a lot of Java classes when they are referenced for the first time. Please be patient and do not start to click on arbitrary buttons to get some response from the system.

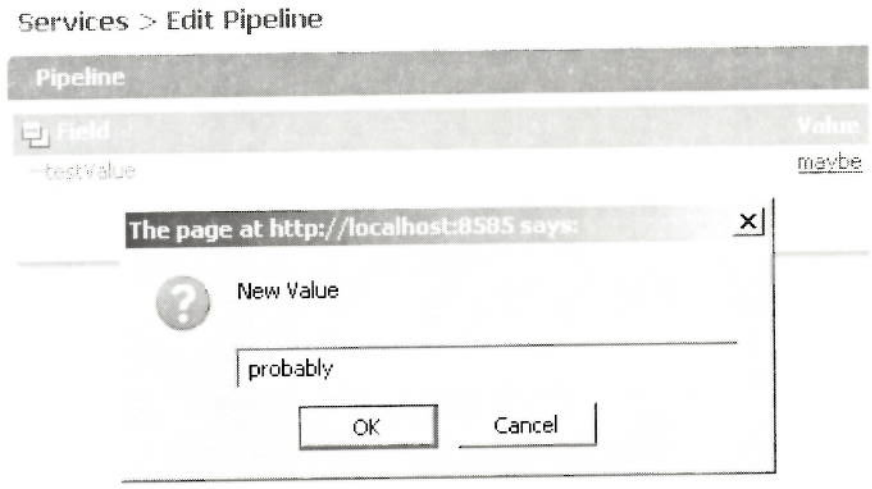
   In the navigation bar on the left side select Applications ➡ Monitoring ➡ Integration ➡ Services.

   Find the acme.PurchaseOrder.work:branch1 service in the **Services** result list.

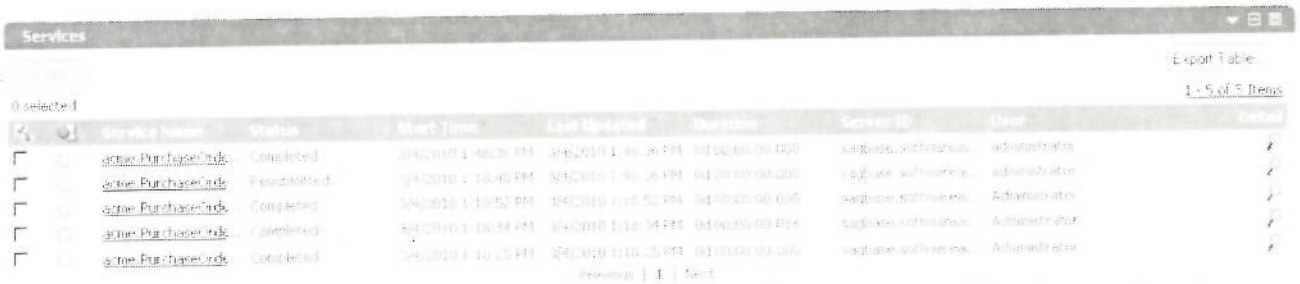


Click on the **View Details** button ( ). The statistics about this individual service execution will be displayed.

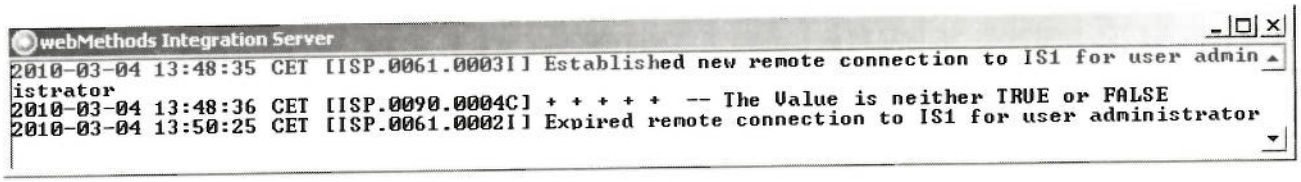
8. Use the **"Edit Pipeline"** button and change the field **testValue**.



Click the OK and **Save** buttons. Finally click on the **Resubmit** button and you will see a "Resubmitted" entry in the **Service Information** tab.



Verify that the service resubmission is also shown in the Server Log file:



Check Your Understanding

1. Why is it necessary to create remote server aliases?
2. Under what circumstances would it be acceptable to resubmit a service? Why?

This page intentionally left blank

Exercise 12:
Invoking Services

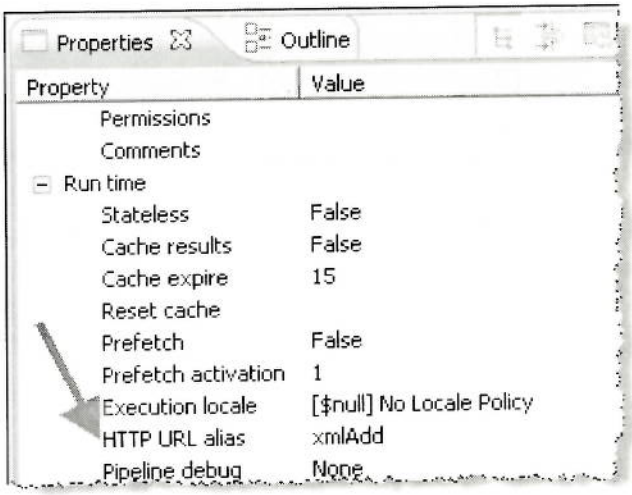
Overview

In this Exercise, you will use different ways to invoke a service.

Steps

1. Invoke a service using HTTP.

   a. To invoke a service using HTTP open designer and find the xmlAdd service in the acmeSupport.xml package. Find it's **"HTTP URL alias"** property and set it to **xmlAdd**.



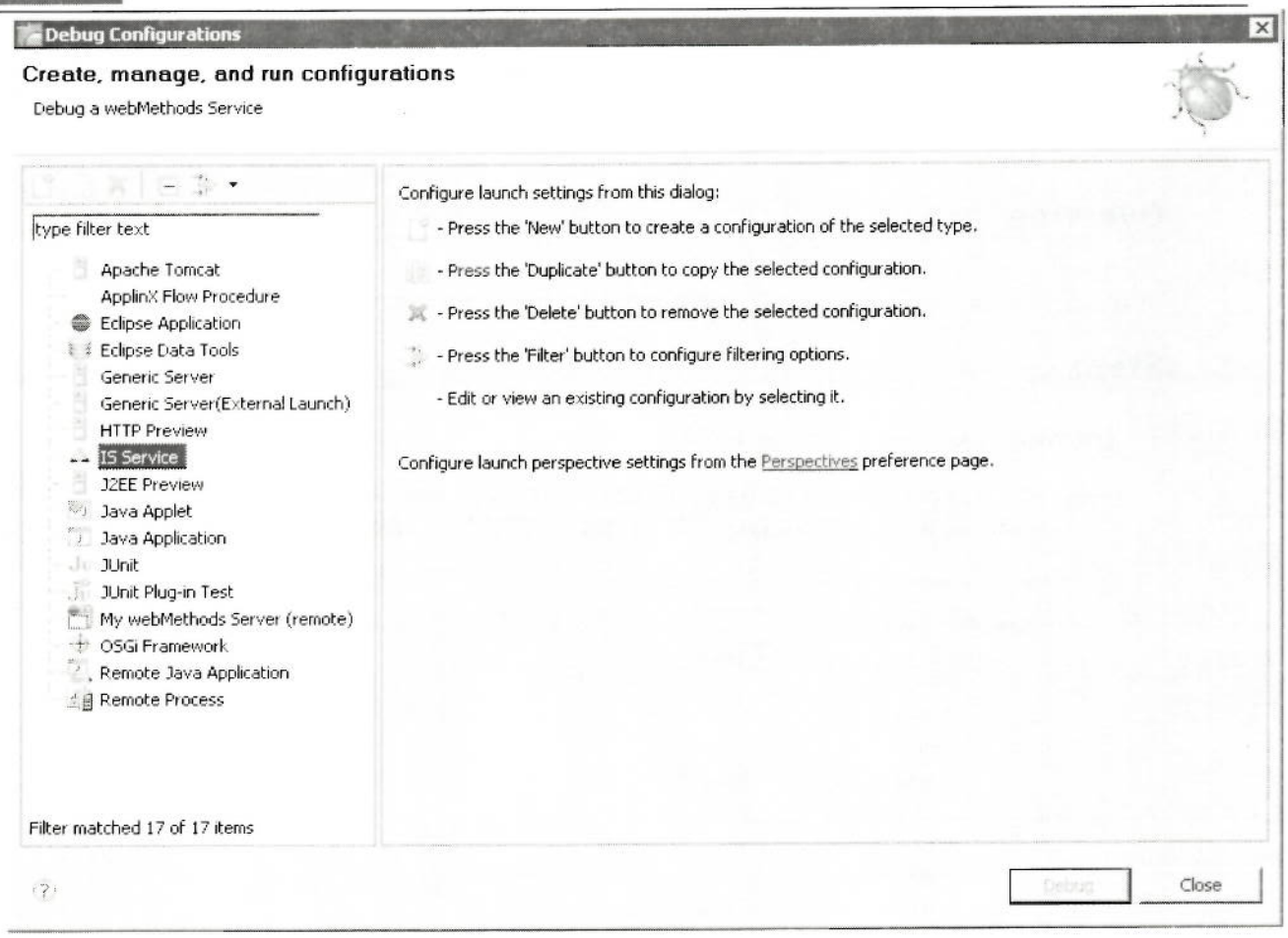
   b. Save the service.

   c. Open a browser and visit the URL **"http://localhost:5555/xmlAdd? a=12&b=23"**. Now open the alternative URL **"http://localhost:5555/invoke/acmeSupport.xml/xmlAdd? a=12&b=23"**. Compare the two results for differences.

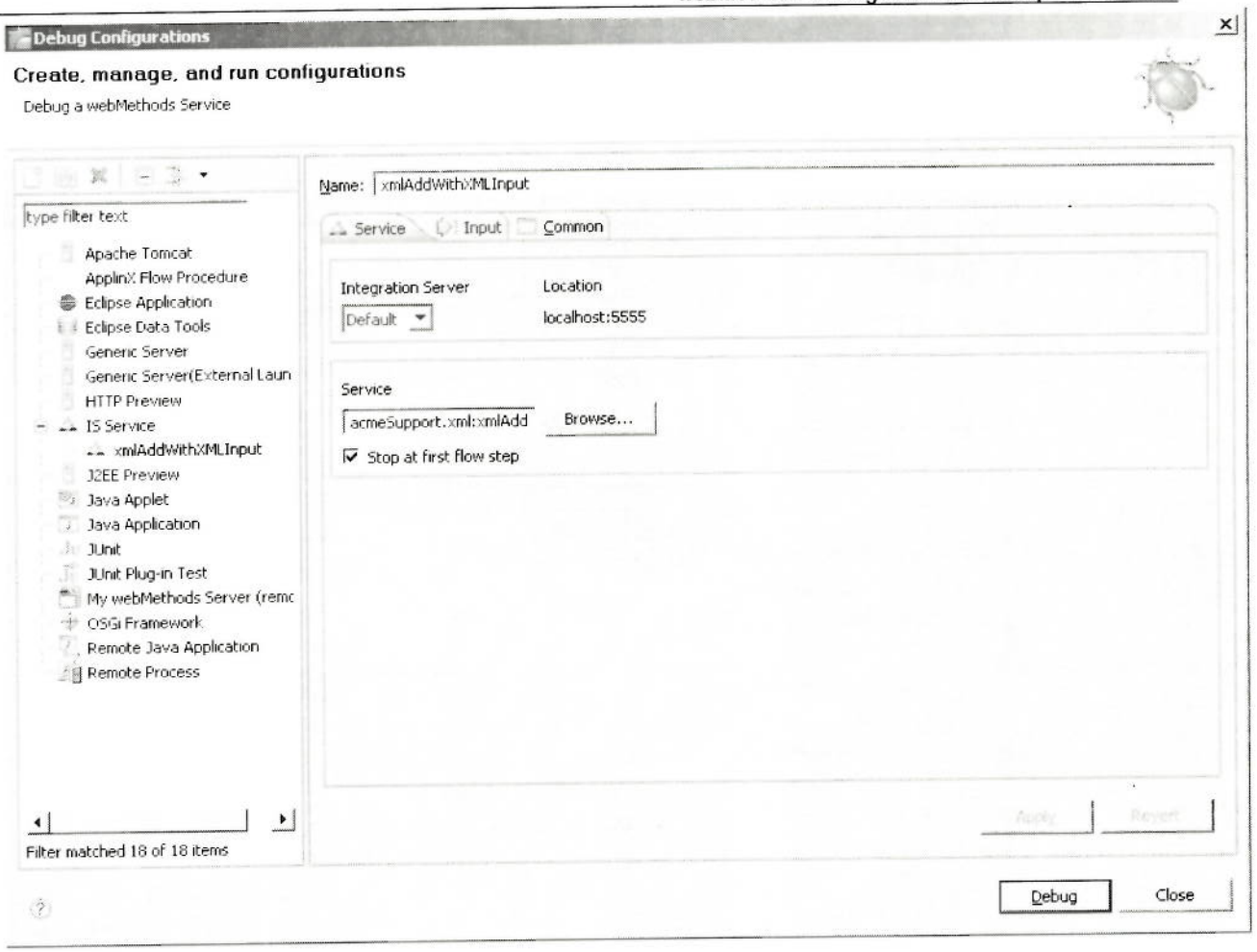
aList	12
bList	23
value	35

2. Invoke a service with XML input

   a. Right click the **acmeSupport.xml:xmlAdd service** and select **Debug as ➡ Debug Configurations**. In the upcoming Dialogue double click on IS Service:

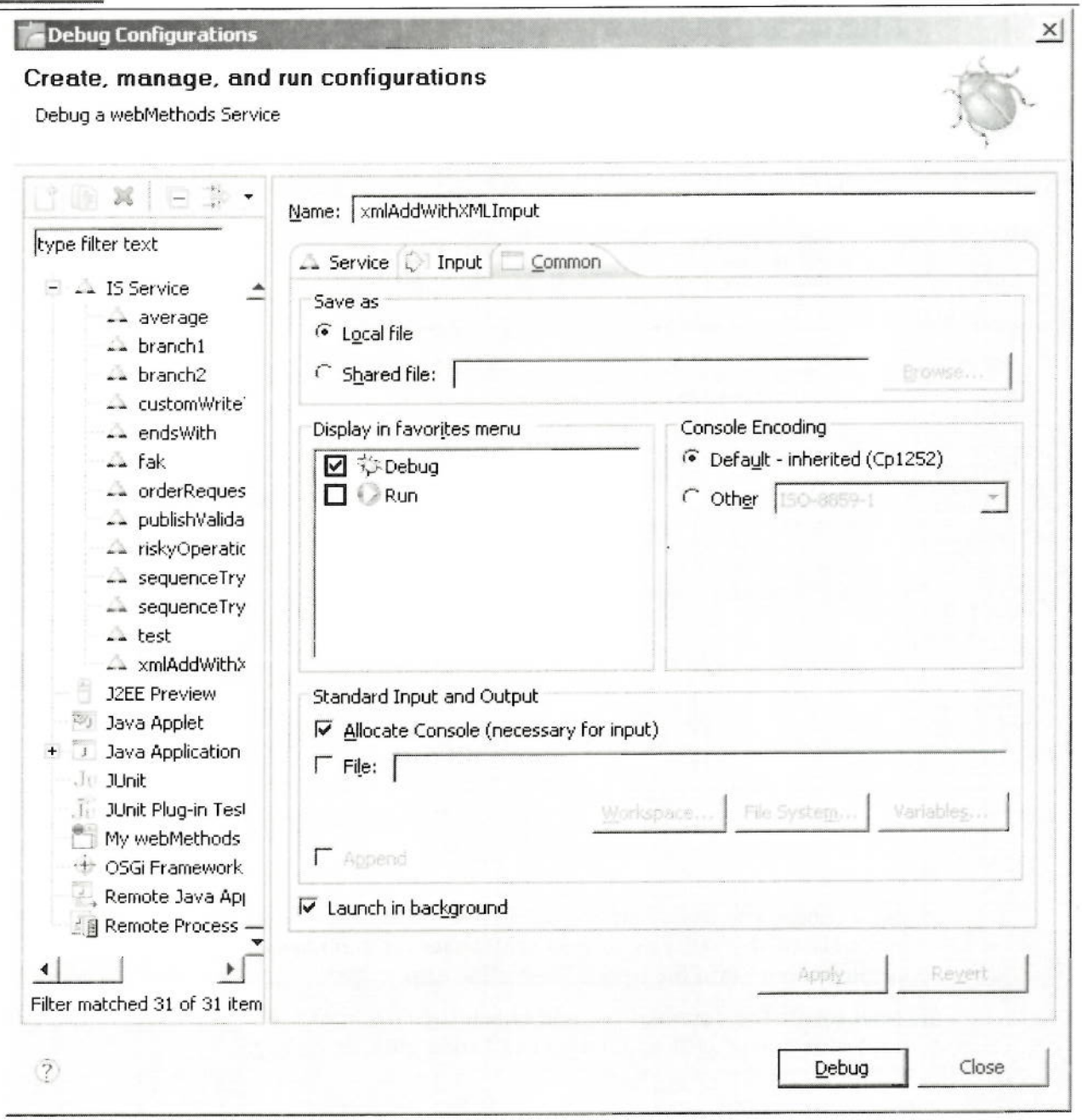


b. Click the browse Button and select the **acmeSupport.xml:xmlAdd** service. Hit OK. Then change the name of your launch configuration from New_configuration to a meaningful name like "**xmlAddWithXMLInput**" and hit Apply.

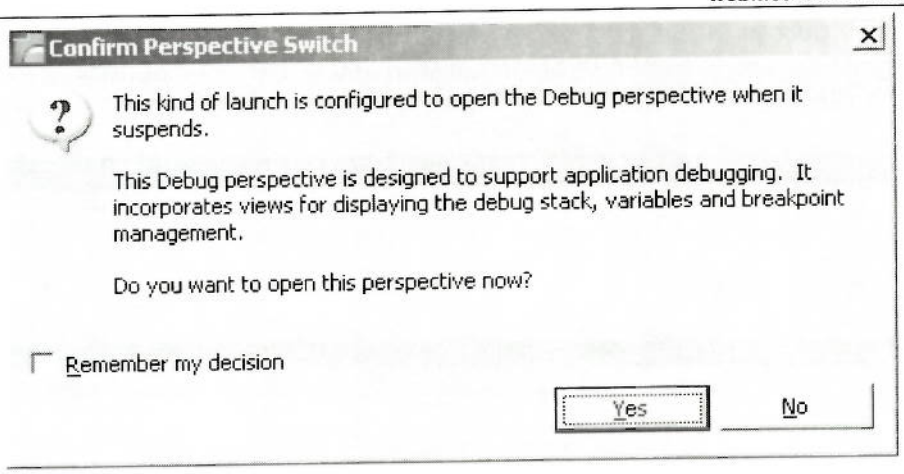


c. Now choose the Input tab and select the "**Use XML**" radio button. Click browse and navigate to the XML File **...\IntegrationServer\packages\AcmeSupport\pub\addInput.xml and hit** Open. Then click **Apply** again.

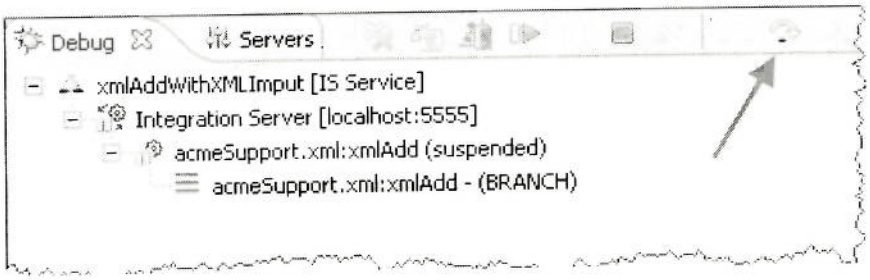
d. Now select the common tab and check the checkbox "Debug" in the "Display in favorites menu". Hit apply again and then click on Debug.



Confirm the dialog asking about a Perspective switch by clicking **Yes**, where you optionally can suppress further appearances of this dioalogue by checking the "Remember my decision" checkbox.

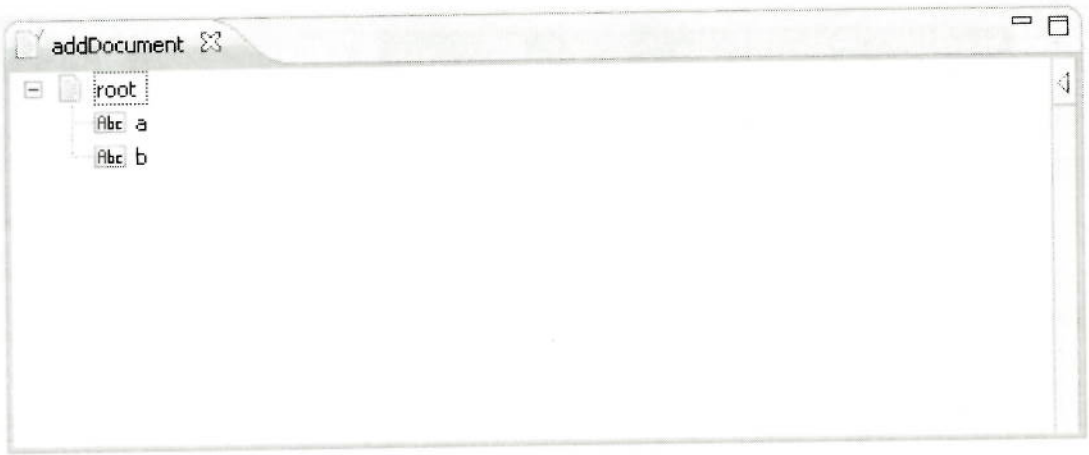


e. Now the debugger comes up and you are debugging the Service with a single input variable node of type Object already in the pipeline. Step through the service to see how this gets converted and added.



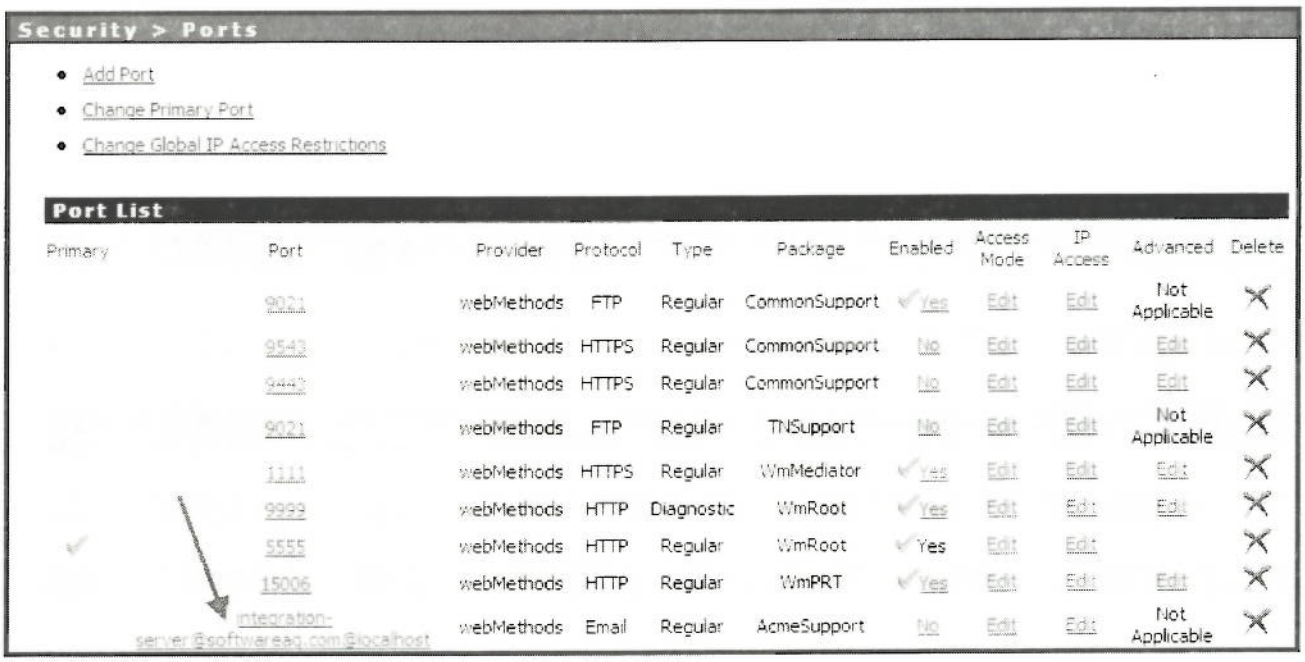
3. Invoke a service using SMTP (mail)

a. Open designer and inspect the document **acmeSupport.xml:addDocument**. You should find that it contains a simple root node containing two variables called a and b.

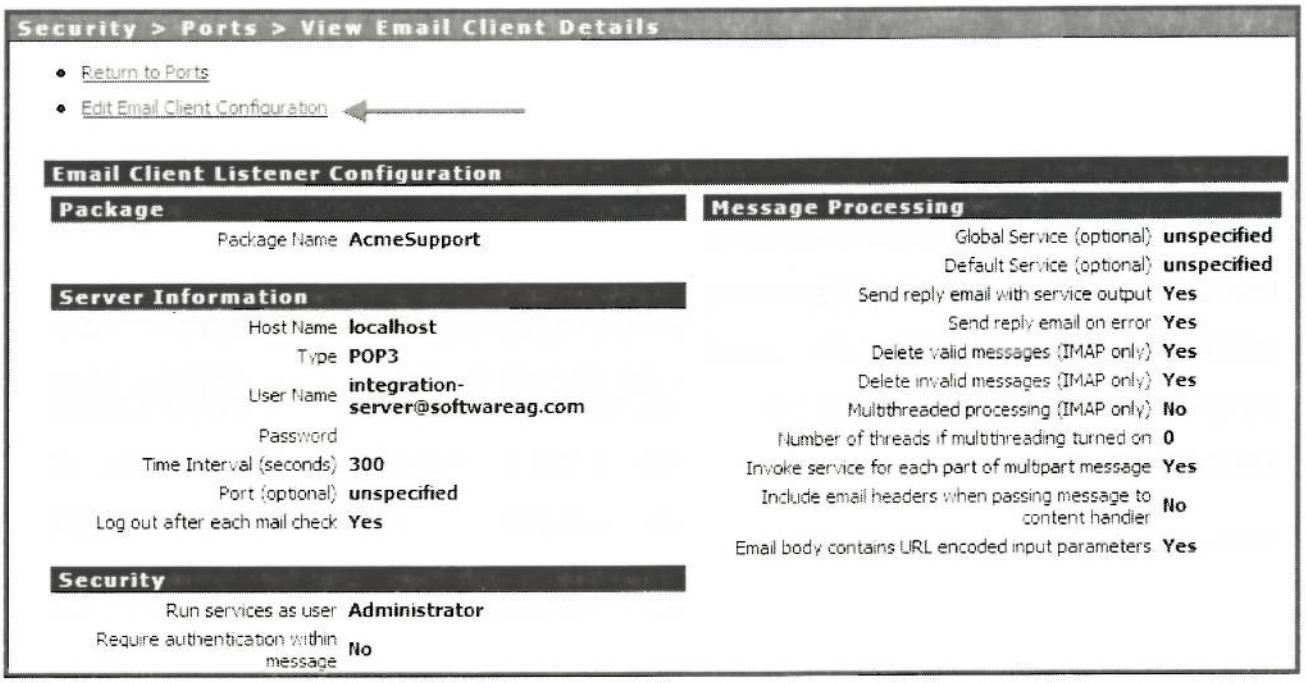


b. Now inspect the xmlAdd service in the same folder and find out what it is doing. Also have a look at the content of the file **...\IntegrationServer\packages\AcmeSupport\pub\addInput.xml**. Given this file as input to the xmlAdd service, what result would you expect?

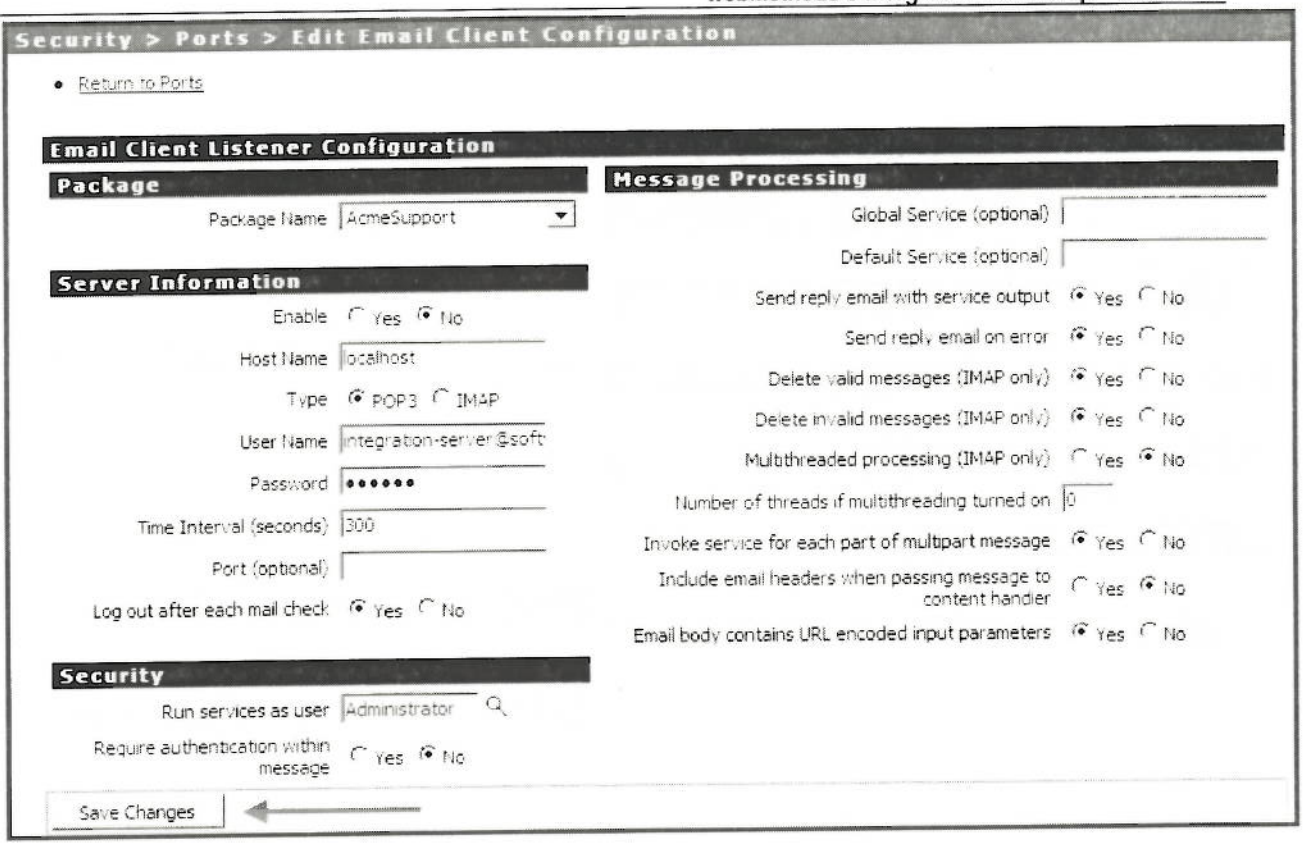
c. Enable the Email port in Integration server, which is turned off by default. To do so, open the Integration server administration console and got to the Security ➡ Ports menu. Click on the integration-server@softwareag.com@localhost entry



and choose edit email client configuration

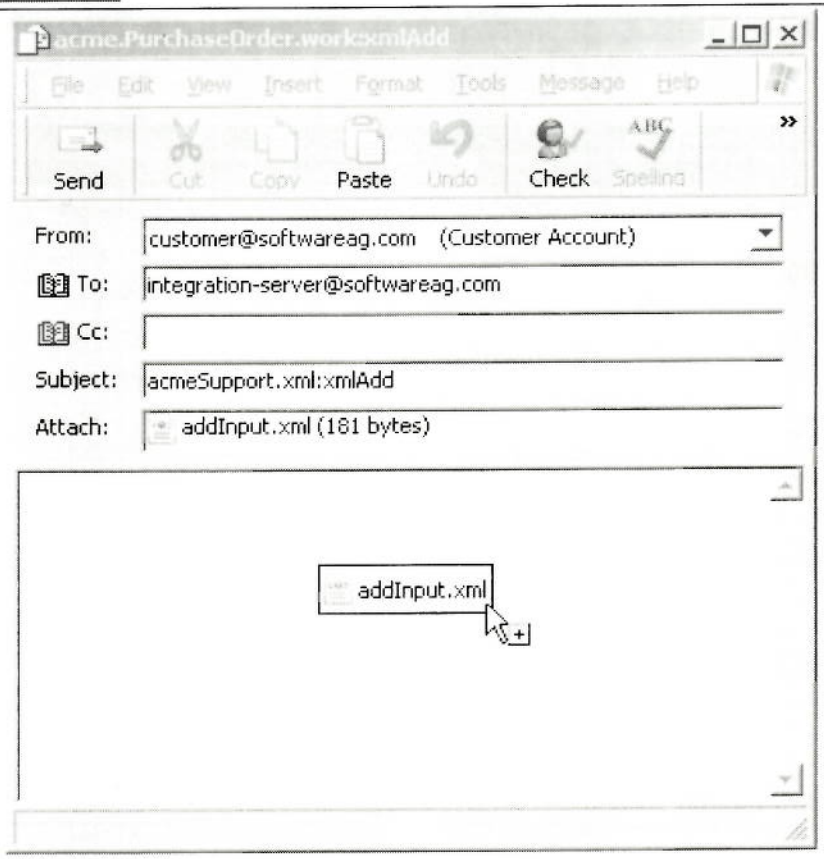


enter the password of **manage** and click save changes

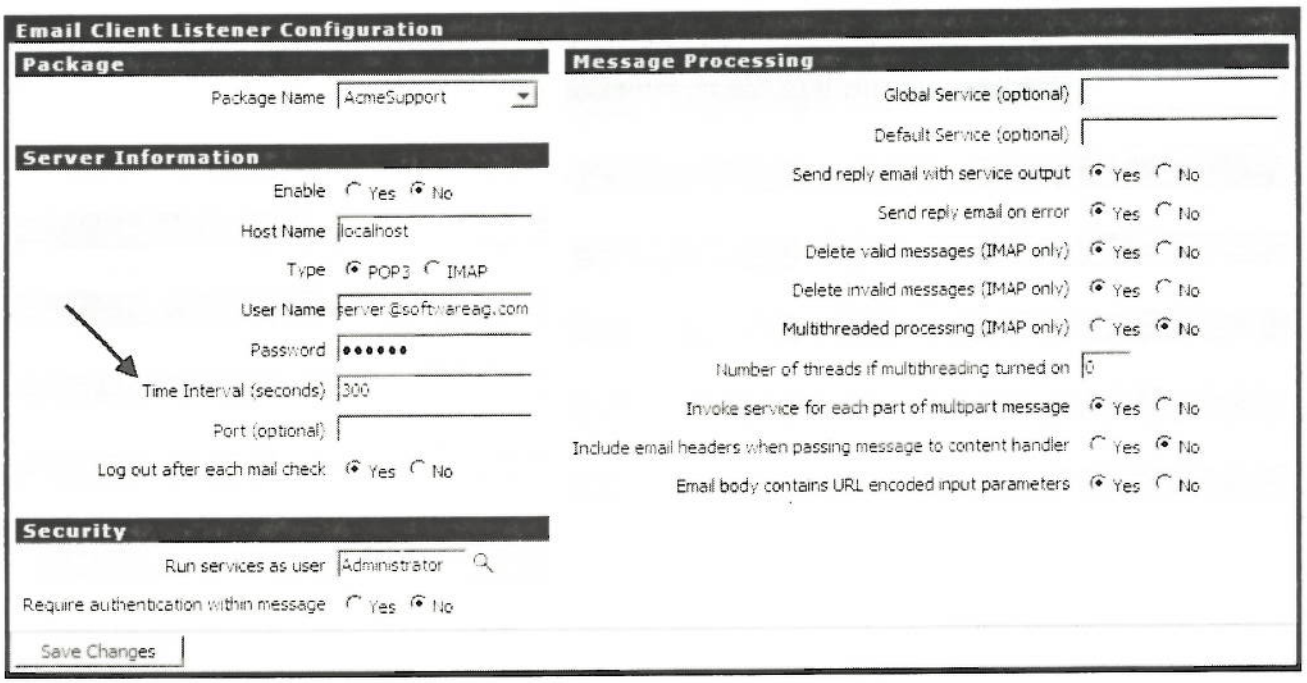


click on the word "No" in the "Enabled" column to activiate this port.

d. Start Microsoft Outlook Express and send a mail from **customer@softwareag.com** to **integration-server@softwareag.com** with an empty message body and the subject set to **acmeSupport.xml:xmlAdd.** As attachment drag the **addInput.xml** document from an explorer window into the mail message.
Once you completed you mail, press the send button.



Note1: In order speed up processing, you may want to change the parameter "Time Interval (seconds) from the configured value of 300 to something less like 10 seconds.



Do not forget to enable the Port after changing this value. Remember to change this value back to 300 after the exercise.

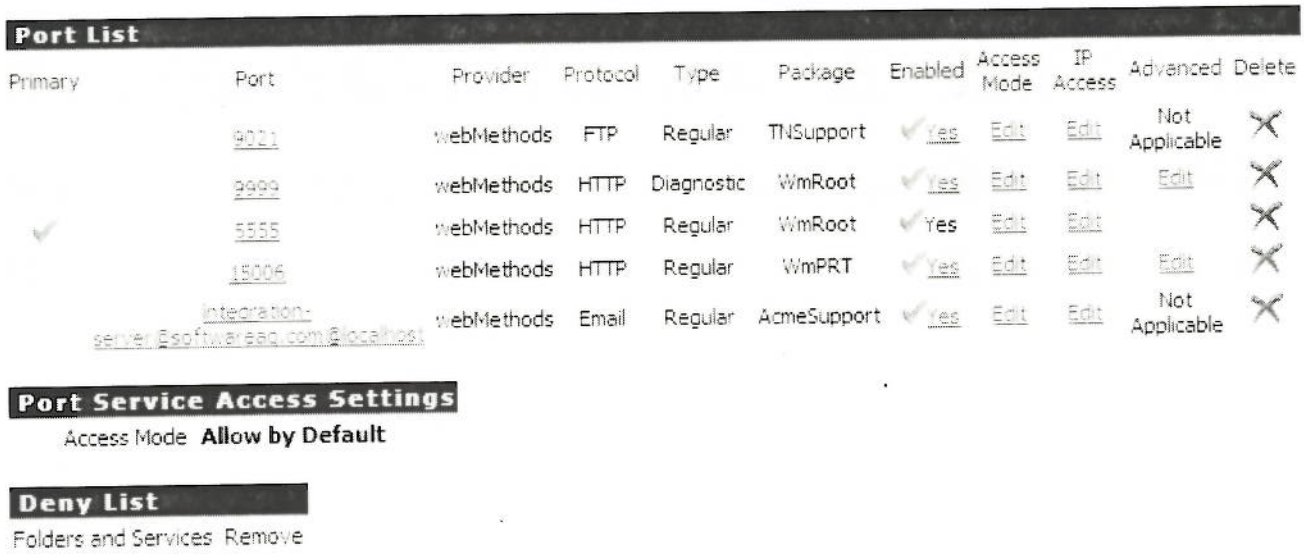
Note2: The mail service and the outlook express program on your virtual machine are set up to handle all mail locally. There is no connectivity to any outside mail system.

Outlook Express and the hmail server are set up to serve the softwareag.com and the v8training.net domains. Please do not change any of the configuration settings unless otherwise noted.

After you sent your mail, press the Send/Recv button in Outlook Express and you will receive a reply from integration server with the result of the message processing. To see the content of this message, simply drag it into the window of a running onstance of the Notepad++ editor.

4. Invoke a service using FTP.

a. Before using ftp, make sure there is an enabled FTP port in integration server available. Open the Integration Server administration tool and go into the security ➡ ports submenu. Make sure an FTP port exists for port 9021 and make sure its access mode setting allows every service to be executed:



Port List

Primary	Port	Provider	Protocol	Type	Package	Enabled	Access Mode	IP Access	Advanced	Delete
	9021	webMethods	FTP	Regular	TNSupport	Yes	Edit	Edit	Not Applicable	
	9999	webMethods	HTTP	Diagnostic	WmRoot	Yes	Edit	Edit	Edit	
✓	5555	webMethods	HTTP	Regular	WmRoot	Yes	Edit	Edit		
	15006	webMethods	HTTP	Regular	WmPRT	Yes	Edit	Edit	Edit	
	Integration-server@softwareag.com@localhost	webMethods	Email	Regular	AcmeSupport	Yes	Edit	Edit	Not Applicable	

Port Service Access Settings

Access Mode **Allow by Default**

Deny List

Folders and Services Remove

b. Now open a windows command promt window and execute the command script as shown below. Your input is shown **in bold font**. Make sure you understand what each command is doing before typing it in.

```
C:\>cd /d C:\SoftwareAG\IntegrationServer\packages\AcmeSupport\pub

C:\SoftwareAG\IntegrationServer\packages\AcmeSupport\pub>dir addInput.xml
 Volume in drive C has no label.
 Volume Serial Number is 9C80-4210
 Directory of C:\SoftwareAG\IntegrationServer\packages\AcmeSupport\pub

03/09/2010  03:54 PM               181 addInput.xml
               1 File(s)            181 bytes
               0 Dir(s)  42,108,518,400 bytes free

C:\SoftwareAG\IntegrationServer\packages\AcmeSupport\pub>ftp
ftp> open localhost 9021
Connected to sagbase.softwareag.com.
220 sagbase:9021 FTP server (webMethods Integration Server version 8.0.1.0)
ready.
User (sagbase.softwareag.com:(none)): Administrator
331 Password required for Administrator.
Password: manage
230 User Administrator logged in.
ftp> cd ns
250 CWD command successful.
ftp> cd acmeSupport
```

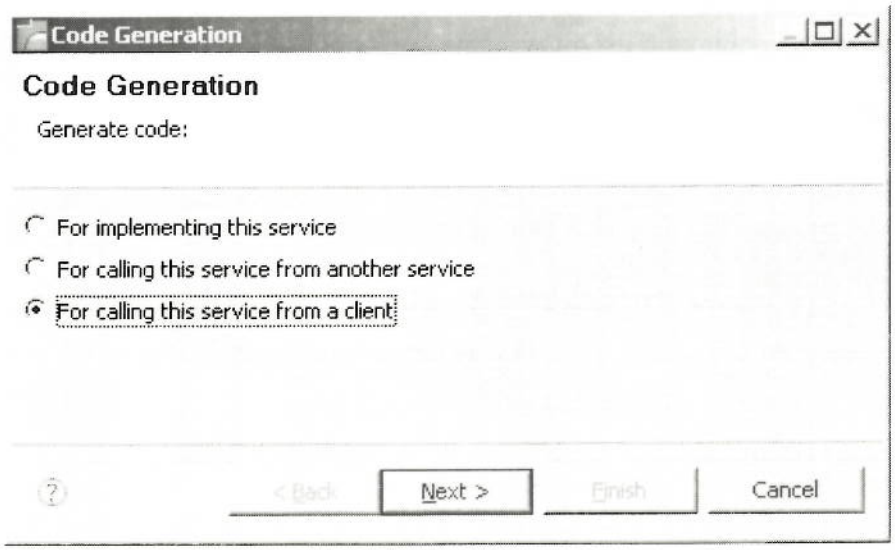
```
250 CWD command successful.
ftp> cd xml
250 CWD command successful.
ftp> cd xmlAdd
250 CWD command successful.
ftp> send addInput.xml
200 PORT command successful.
150 ASCII mode  data connection for addInput.xml (127.0.0.1,2366).
226 ASCII transfer complete.
ftp: 181 bytes sent in 0.00Seconds 181000.00Kbytes/sec.
ftp> dir
200 PORT command successful.
150 ASCII mode  data connection for /bin/ls (127.0.0.1,2368).
total 1
dr-xr-xr-x  3 root      root                        1 Mar 09 16:44 .
dr-xr-xr-x  3 root      root                        1 Mar 09 16:44 ..
-r--r--r--  1 tx        tx                        106 Mar 09 16:44
addInput.xml.out
226 ASCII transfer complete.
ftp: 232 bytes received in 0.02Seconds 14.50Kbytes/sec.
ftp> get addInput.xml.out
200 PORT command successful.
150 ASCII mode  data connection for addInput.xml.out (127.0.0.1,2370) (106
bytes).
226 ASCII transfer complete.
ftp: 106 bytes received in 0.00Seconds 106000.00Kbytes/sec.
ftp>!type addInput.xml.out
<?xml version="1.0" encoding="UTF-8"?>

<Values version="2.0">
  <value name="value">42</value>
</Values>
ftp> quit
221 Goodbye.
```

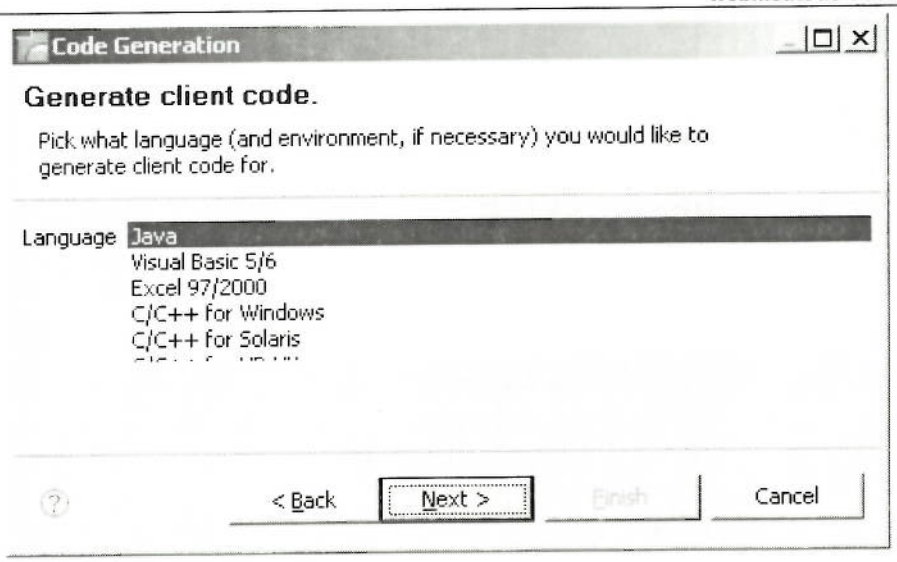
5. Invoke a service using Java

a. Find the xmlAdd service above in Designer.

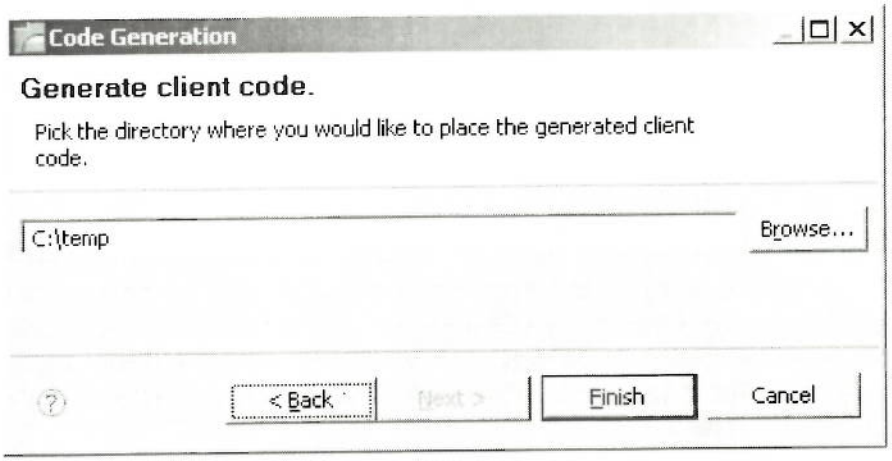
b. Right click the service and choose the "Generate Code" entry. In the dialog box that opens select "For calling this service from a client".



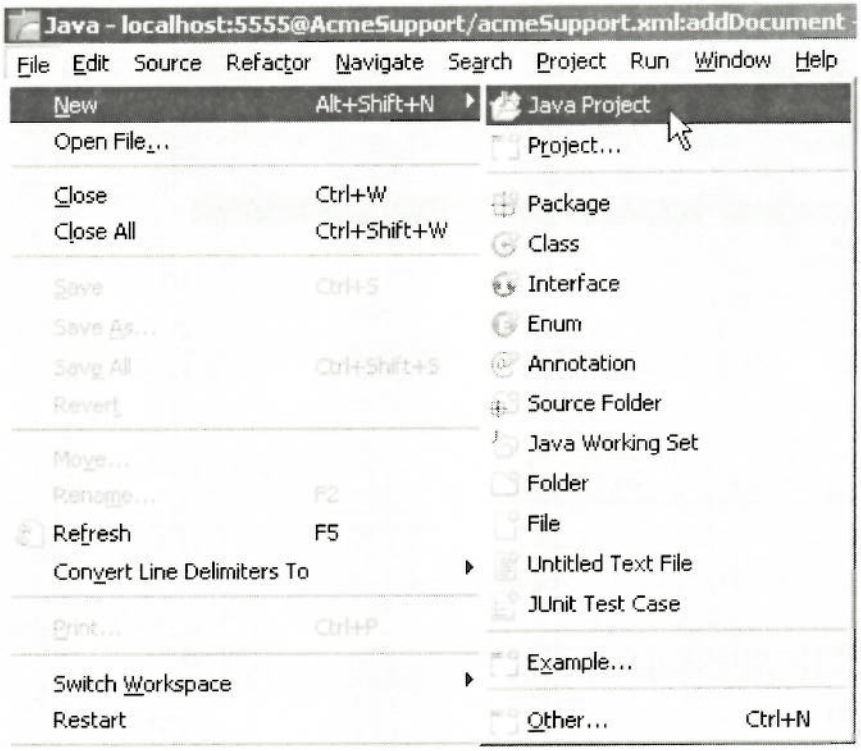
c. Then choose **Java** as language



and use C:\TEMP as directory for code generation.

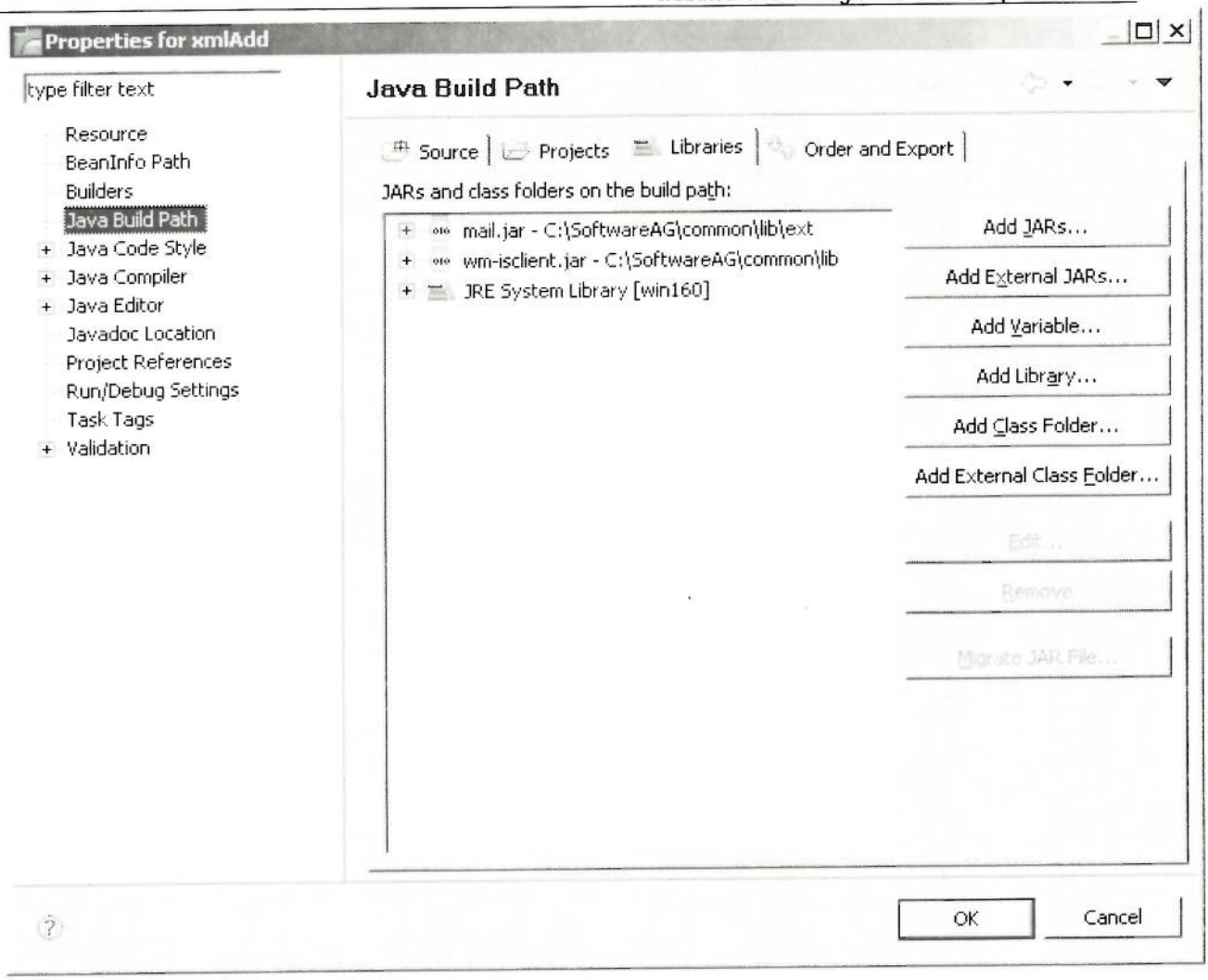


d. Now create an Eclipse Java project.



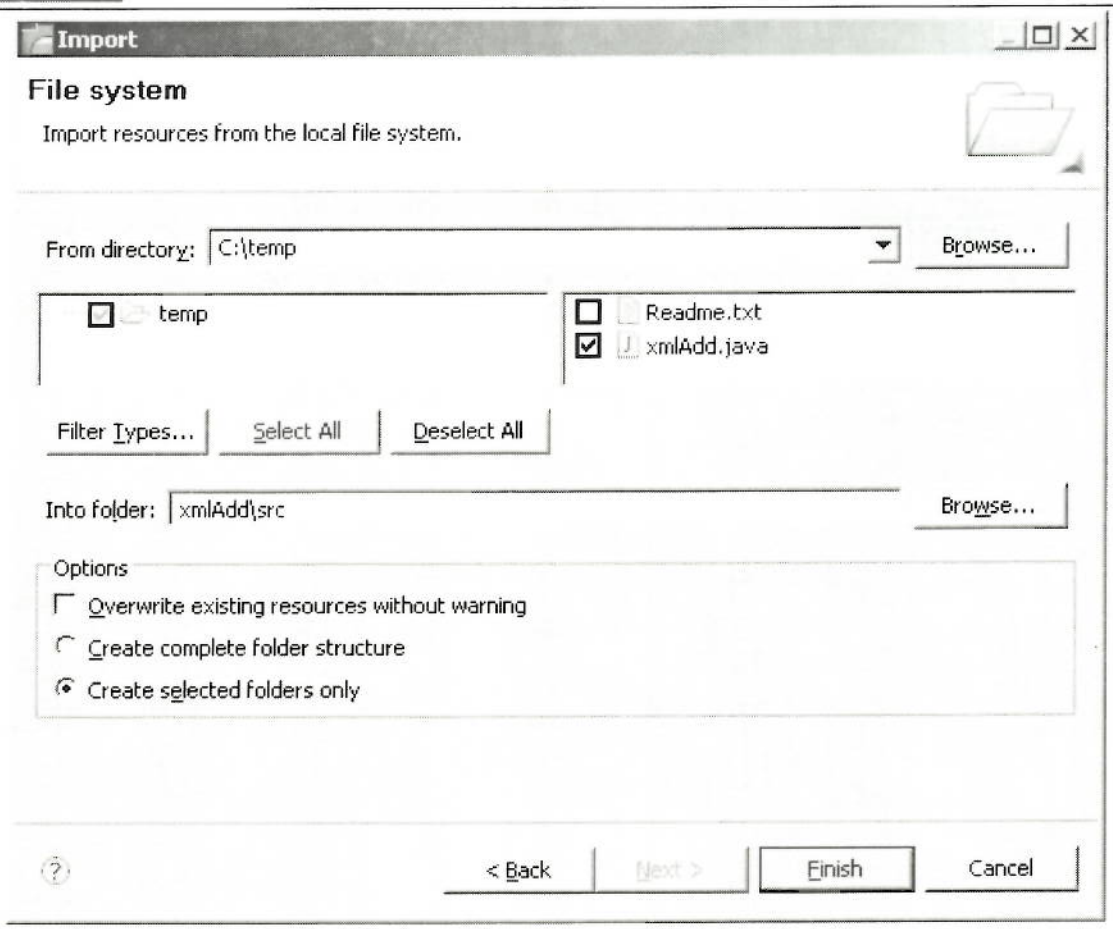
In the upcoming Dialog enter **xmlAdd** as project name. Do not change any of the defaults and hit the "Finish" button.

e. This Project requires two additional external Jar files. To add them, Right click on the project node (xmlAdd) and select the **Properties** entry at the very bottom of the pop up. In the appearing dialog select "**Java Build Path**" and click on the "**Libraries**" selector. In this window choose "**Add external Jars**" and add (in 2 steps) the Libraries **...\common\lib\wm-isclient.jar** and **...\common\lib\ext\mail.jar**. When finished, your window should look like this one:



Close the dialog by clicking the OK button.

f. Now import the Java source generated in the first step. To do so, right click the xmlAdd node once more and select the "**Import**" option from the menue. Choose "**General**" ➡ "**File System**" and click "**Next**". In the upcoming window enter C:\temp as directory and make sure the xmlAdd.java file is selected. In to "**Into Folder**" field enter "**xmlAdd\src**". Your dialog should look like this:



Click the "**Finish**" button.

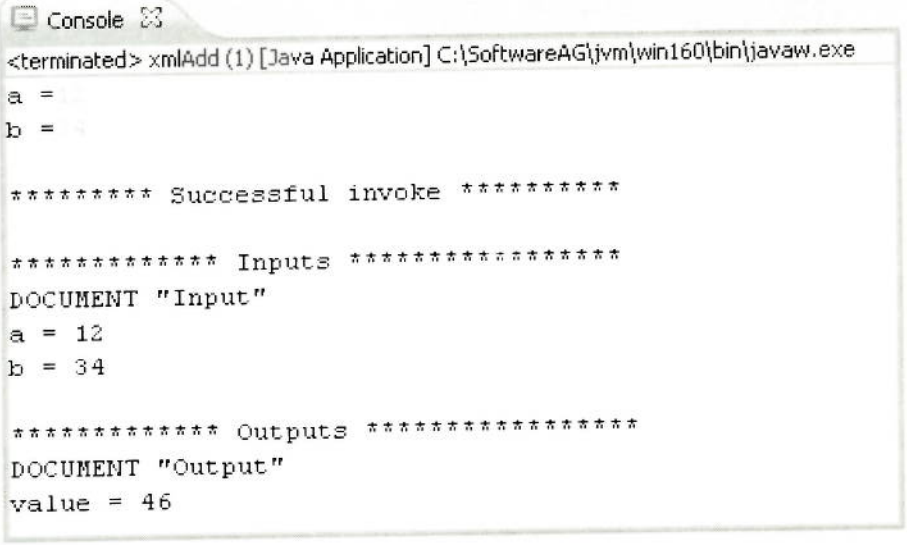
g. Open the xmlAdd.java file and change the lines

```
// Set user name and password for protected services
String username = null;
String password = null;
```

to correct credentials like the following:

```
// Set user name and password for protected services
String username = "Administrator";
String password = "manage";
```

h. Now you can run your program by right clicking the xmladd.java file and selecting "**Run As**" ➡ "**Java Application**". You may have to confirm saving your sources. Look for the console view. Enter two small numbers for the "a =" and "b =" prompts and verify the result.



Check Your Understanding

1. Why and when would you use an HTTP URL alias for your services?
2. How do the services find their input data?
3. How do the services return their result?

This page intentionally left blank

Exercise 13:
Create a Flat File Schema

Overview

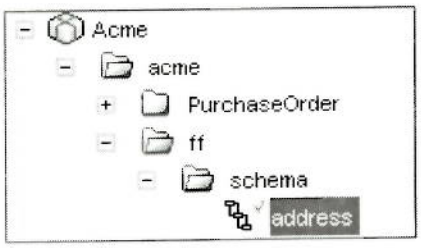
In this exercise, you will create, configure and test a Flat File Schema object to parse flat files like the following one:

```
ADDRESS,Acme Hammer Company,123 Wilson St.,Sacramento+CA+95833
ADDRESS,Johnson Supply Co.,456 Nadia Ave.,Seattle+WA+98188
```

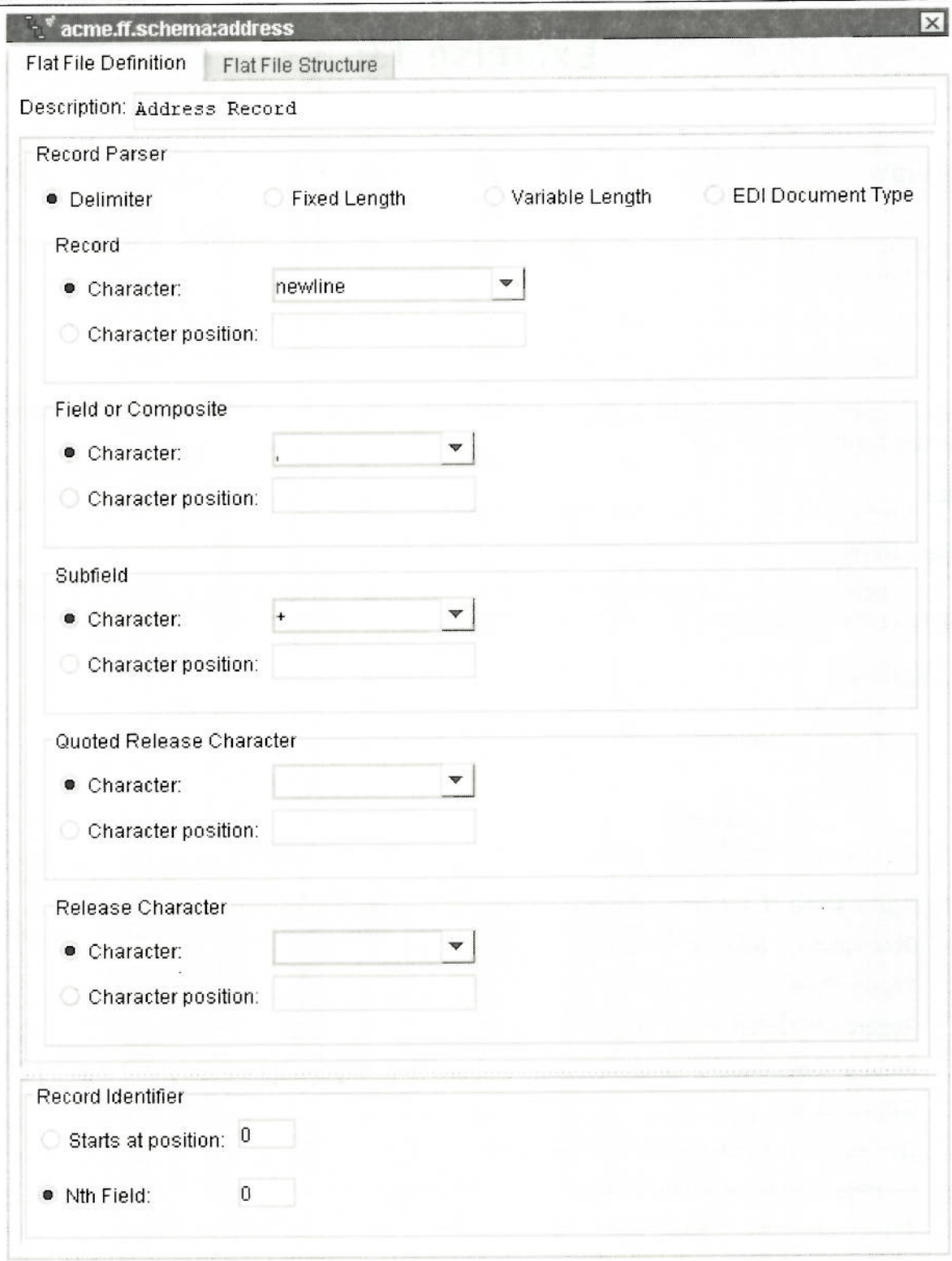
However, since the flat file processing is not jet part of designer, you have to use the developer tool to perform this exercise.

Steps

1. Start Developer
2. In the **Acme** package's **acme** folder, create a new folder called **ff** and a folder inside of **ff** called **schema**. Create a new **Flat File Schema** object called **acme.ff.schema:address.**

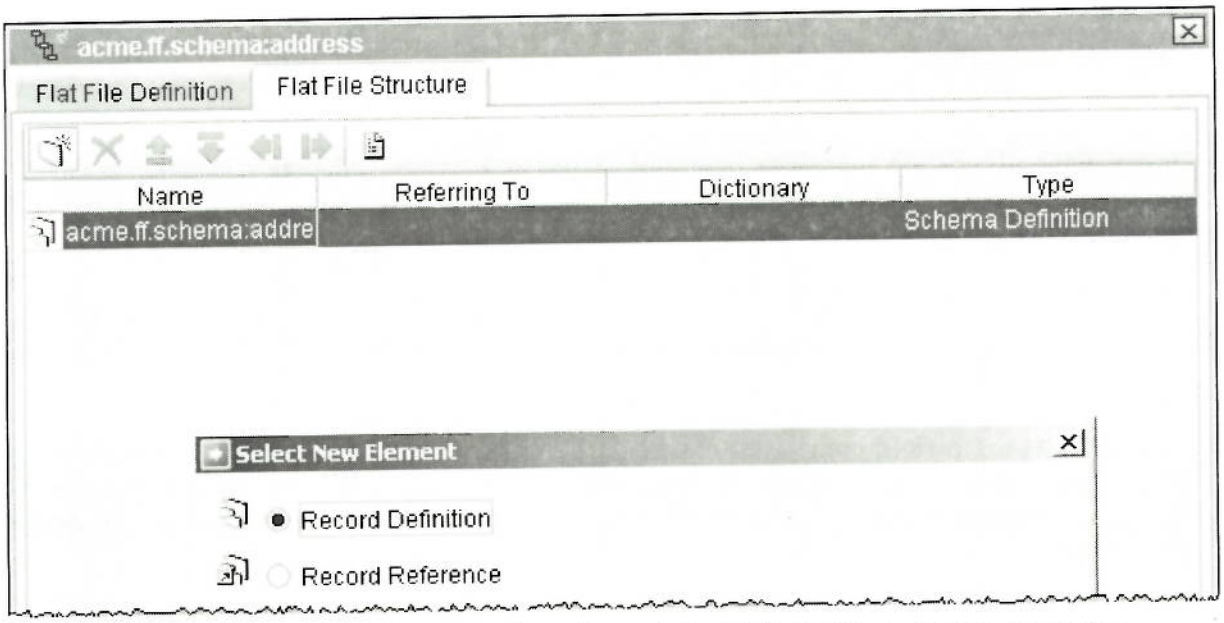


3. Configure **acme.ff.schema:address** to have the following information:
   a. Description = Address Record
   b. Record Parser = Delimiter
   c. Record Character = newline
   d. Field or Composite Character = ,
   e. Subfield Character = +
   f. Quoted Release Character = *leave blank*
   g. Release Character = *leave blank*
   h. Record Identifier = Nth Field: 0

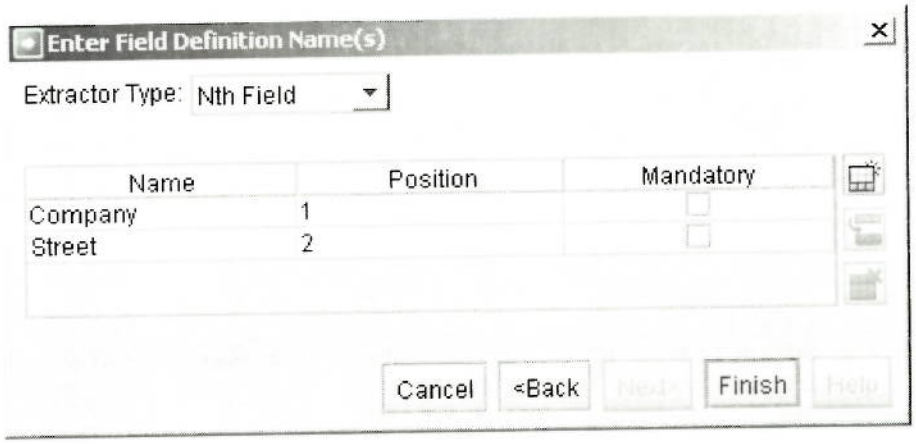


Note: due top limited screen space, you may have to move some of the sliders and scrollbars to be able to see all the input fields.

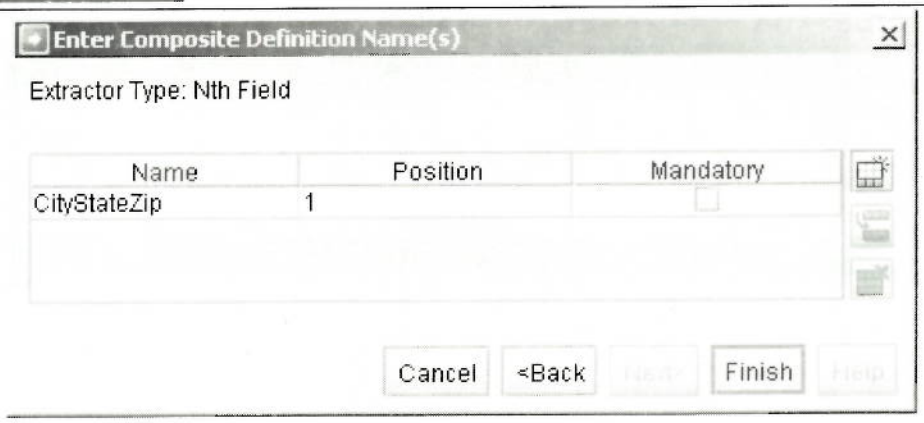
4. Change to the **Flat File Structure** tab. Select the **address** schema with the mouse and create a record definition (by clicking on the  button) called **ADDRESS**.



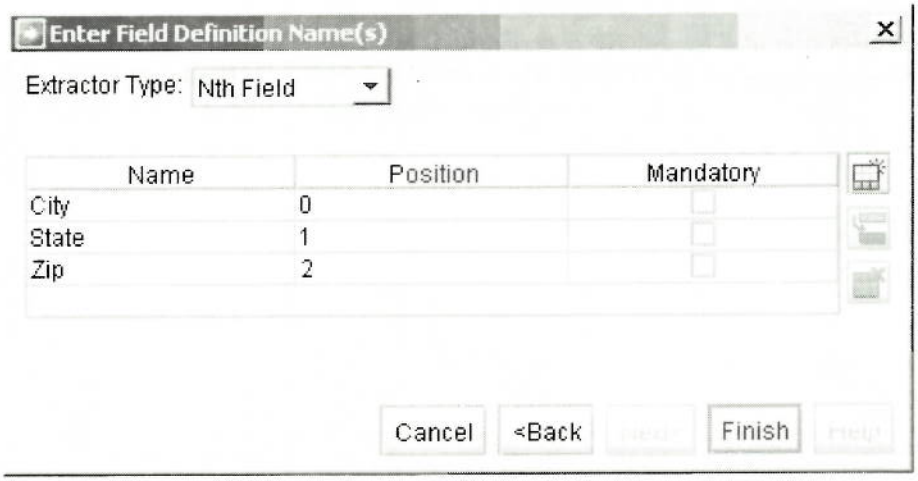
5. Right click the **ADDRESS** record and select **New ➡ Field Definition**. In the upcoming dialogue select an **Extractor Type** of **Nth Field**. Then create 2 **Field Definitions**, called **Company** and **Street** at positions 1 and 2, respectively. Note: make sure you select the extractor type before entering the field names.



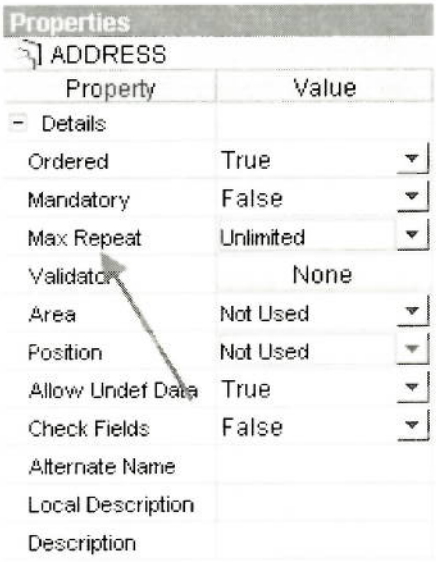
6. Now create a Composite Definition called **CityStateZip** at position 3. This is done by closing the above dialogue and right clicking the **ADDRESS** record definition. Choose **New ➡ Composite Definition** and fill in the Name Field with **CityStateZip.** The Position Field gets the number 1 assigned.



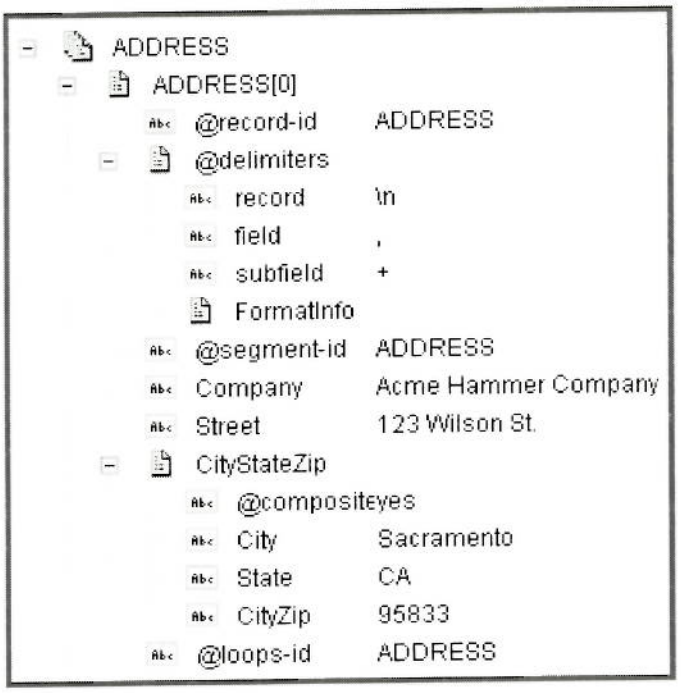
7. In the **CityStateZip** field composite create 3 subfields. They are created by right clicking the composite field and choosing **new ➡ Field Definition**. Make sure an extractor of type Nth Field is used and call your new fields **City**, **State**, and **Zip** and assign positions 0, 1, and 2, respectively.



8. Set the **ADDRESS** record definition to have an **Unlimited** value for the **Max Repeat** property.



9. Save and test the new **Flat File Schema** called **address** by running it in Developer. When asked for an input file, use **address.txt** in the directory **...\IntegrationServer\packages\AcmeSupport\pub\FlatFile.** Make sure that the first ADDRESS record looks like the following:



Check Your Understanding

1. Why can't flat files be imported like XML documents? There not structure and the do not have schemas that is provided with flat file
2. What is the meaning of Nth field?

This page intentionally left blank

Exercise 14:
Create a Flat File Dictionary

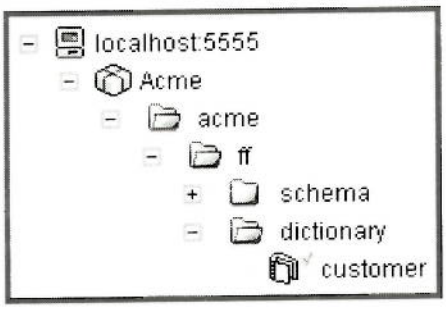
Overview

In this exercise, you will create a new Flat File Dictionary, create a reusable record definition in the Dictionary, reference this record definition in a new Flat File Schema, and test the Flat File Schema.

Just like the last exercise, you have to do this exercise using the developer tool.

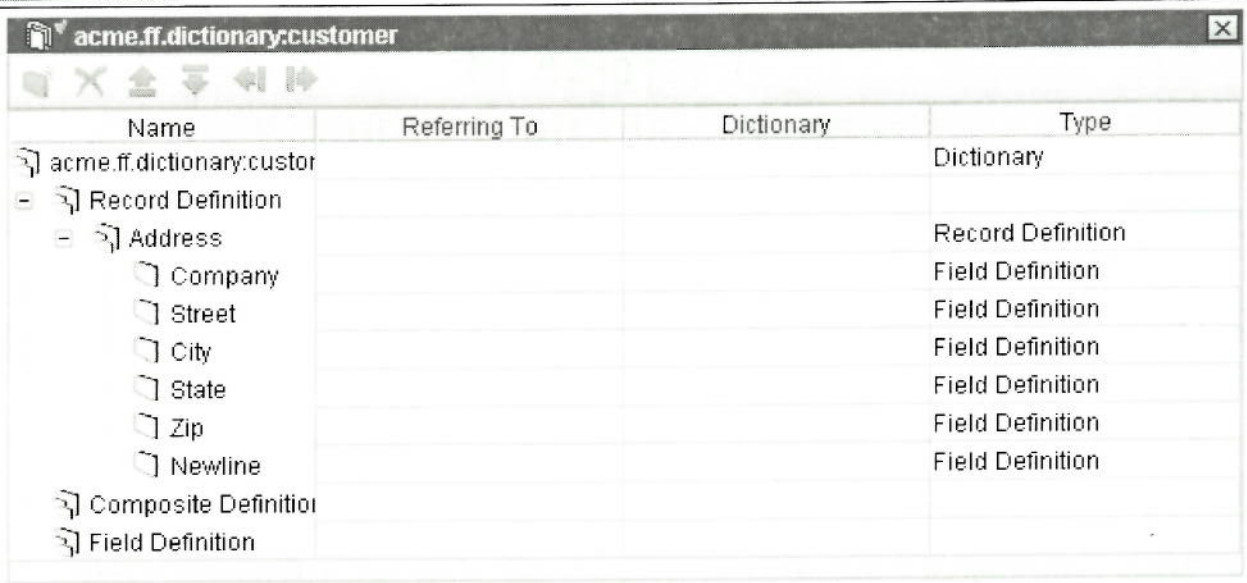
Steps

1. In the Acme package's **acme.ff** folder, create a new folder called **dictionary**. Create a **Flat File Dictionary** called **acme.ff.dictionary:customer.**

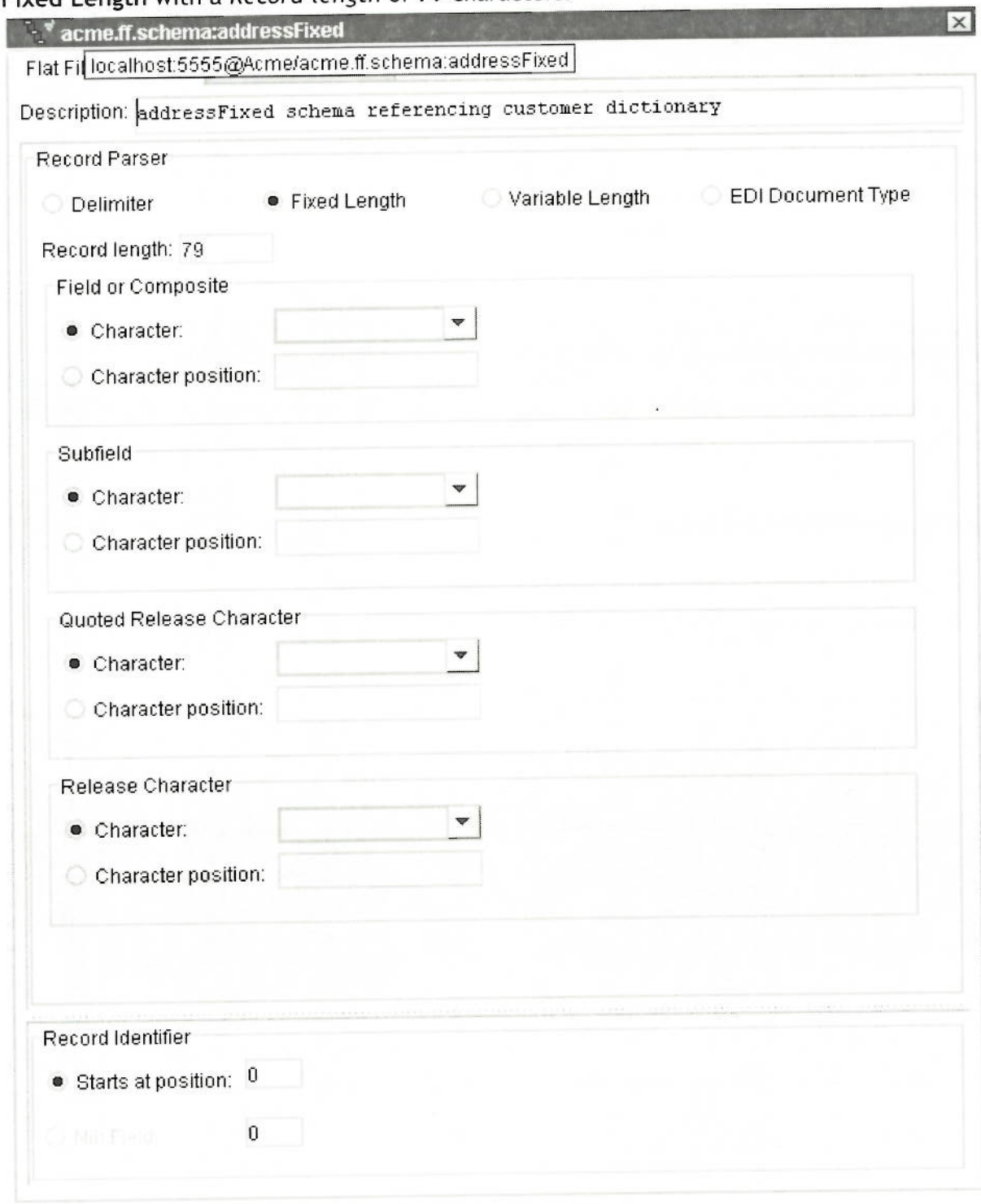


2. Add a record definition called **Address** (case sensitive) to the new **Flat File Dictionary**. Using an **Extractor Type** of **Fixed Position**, add 6 new field definitions to the **Address** record as shown in the table below:

Name	Start	End
Company	0	30
Street	30	55
City	55	70
State	70	72
Zip	72	77
Newline	77	79



3. Create a new **Flat File Schema** called **acme.ff.schema:addressFixed**. Specify that it is **Fixed Length** with a *Record length* of **79** characters.



4. In the **"Default Record"** property of the **addressFixed** schema, add a reference to the **customer** Flat File Dictionary's **Address** record definition.

Properties

acme.ff.schema:addressFixed

Property	Value
− Default Record	
Set	
Delete	Delete
Dictionary	acme.ff.dictionary:customer
Name	Address

5. Save your work and test the **addressFixed** Flat File Schema with the file **...\ IntegrationServer\packages\AcmeSupport\pub\FlatFile\addressFixed.txt**
Once the **addressFixed** schema functions correctly, click on the **Flat File Structure** tab and select the **Create Document Type** icon ( ) to create the **addressFixedDT** IS document type.

Check Your Understanding

1. What is the difference between a dictionary and a schema?
2. Why should you create the IS document type when the schema is complete?

Flat File Schema under Developer helpdoc.

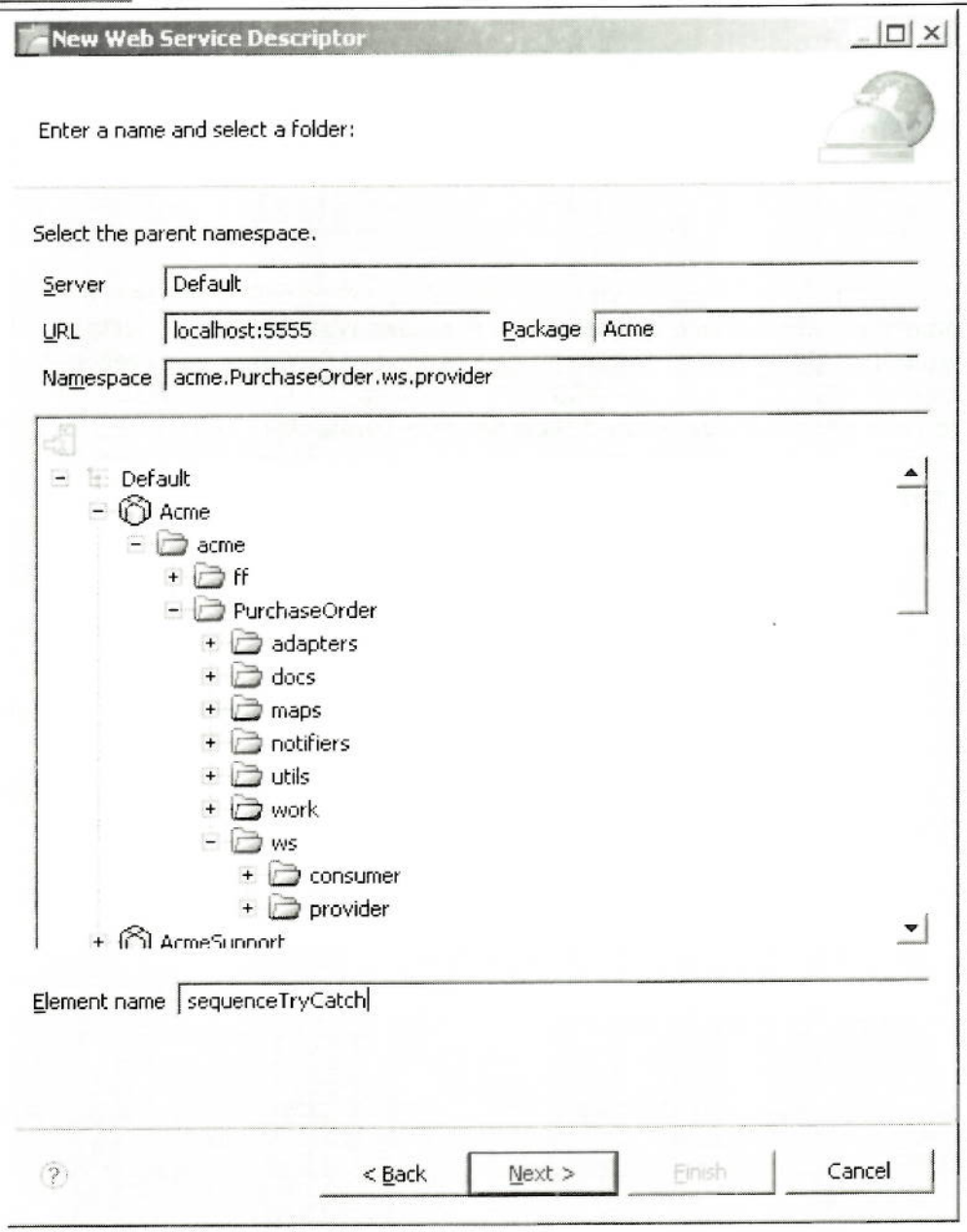
Exercise 15:
Web Service Descriptors and Custom Faults

Overview

In this exercise, you will take the flow service you already created called **sequenceTryCatch** and make it callable via a web service by creating a **Provider Web Service Descriptor (WSD)**. To prove that anyone (including the IS itself) can call **sequenceTryCatch** as a web service, you will create a **Consumer WSD** based on the **WSDL** created from the **Provider WSD** and invoke **sequenceTryCatch** using the auto-generated **Web Service Connector**. Finally, you will create a generic **Error** document. You will specify that it can serve as a custom SOAP Fault. Then you test to see if the custom SOAP fault gets returned as expected.

Steps

1. In Designer, create a new **Web Service Descriptor** in the **acme.PurchaseOrder.ws.provider** folder.
   a. Accept all of the defaults (**Provider**, **Existing IS service(s)**, and **No** for WS-I compliance) and click the **Next>** button.
   b. Type the name **sequenceTryCatch** and specify the **acme.PurchaseOrder.ws.provider** folder as the location to create the **Provider WSD**, then click the **Next>** button.



c. In the dialog that appears next, navigate to and select the **acme.PurchaseOrder.work:sequenceTryCatch** flow service, then click the **Next>** button.

d. In the next screen, leave all the defaults in place and click the **Finish** button.