IBM WEBSPHERE 5.0 Skills Transfer - LAB EXERCISE

Developing Web Services - Part 1

What This Exercise is About

WebSphere Studio Application Developer has tools for working with each part of the Web Services programming model. For the Service Provider, there is a wizard for turning applications into Web Services. For the Service Requester, there is a wizard for creating client applications to access Web Services. Finally, for the Service Registry, there is a Web Services explorer for viewing UDDI public or private registries. This exercise will feature some of the Web Service tools available in WebSphere Studio Application Developer.

User Requirement

User must have IBM DB2 Universal Database (version 7.2) and IBM WebSphere Studio Application Developer Version 5.0.1 installed on a Windows 2000 workstation with Service Pack 3. Internet Explorer version 5.5 is also required. The WebSphere Studio Application Developer should be installed at C:\Program Files\IBM\WebSphere Studio. To complete all parts of the lab, the WebSphere Application Server Version 5.0.1 and WebSphere Network Development Version 5.0.1 packages are required. The lab source files (LabFiles50.zip) must be extracted to the root directory (e.g. C:\). Experience with previous versions of WebSphere Studio Application Developer and the J2EE programming model are also required. Throughout this lab, a userid profile named wsdemo with a password of wsdemo1 is assumed to be created on the system.

NOTE: This lab exercise makes the following assumptions: DB2 is installed at C:\SQLLIB; WebSphere Application Server Version 5.0,1 is installed at: C:\WebSphere\AppServer; WebSphere Network Deployment Version 5.0.1 is installed at: C:\WebSphere\DeploymentManager; WebSphere Studio Application Developer is installed at C:\Program Files\IBM\WebSphere Studio; and the LabFiles50 is extracted to C:\. Throughout this lab, batch files are used to simplify some selected operations. If the software is installed in different locations other than those specified above, you can modify the batch files, however, extensive testing has not been completed to verify all possible combinations. To accommodate different installed locations of the required products above, a batch file called setupVARs.bat has been created. It is located in \LabFiles50. You may update this file according to the instructions contained therein - specifying the drive letters and/or paths of the installed products. All subsequent batch files used throughout this lab calls the setupVARs batch file first. Effectively, you have a single location to make updates to product installation drives and directories.

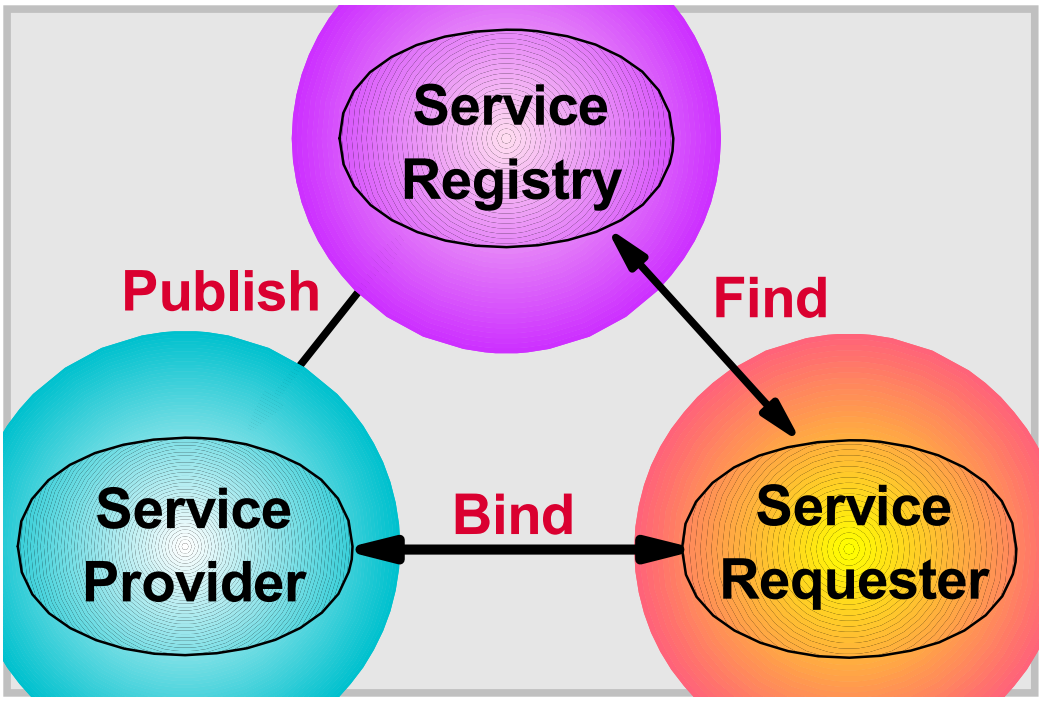
What You Should Be Able to Do

In this lab, the focus is on getting the WebSphere Studio Application Developer workbench environment setup to develop a Web Service. Upon completion of all three parts to this lab exercise, you will have created a Web Service, published it to the Unit Test UDDI Registry and written a sample application to act as a Service consumer.

Background Information

There are three main roles and functions in the Web Services model. The Service Provider provides applications as Web Services and publishes them to the Service Registry. They accept requests, process data, and return the result to Service Requesters. The Service Requester makes these requests (of Service Providers) and receives the result back. The Service Registry is used by both the Service Provider and the Service Requester. The Service Registry lists the different services available by Service Providers and the Service Requester queries the Service Registry for the services available by Service Providers. All of this is done in a loosely-coupled environment as the technical implementation between the different players is based solely on Open Standards (SOAP, XML, WSDL, UDDI), avoiding rigid proprietary implementations. Each role in the Web Services model focuses their efforts on their part of the model, shielded from the other's by the Open Standards.

The following graphic further illustrates the roles and functions in the Web Services model.



Introduction

In Part One, you will complete some necessary steps for the Lab Setup. This lab makes the assumption that you are beginning with the WebSphere configuration files that were backed up after the WebSphere Application Server was installed. This was done in the Installation lab. In addition, the BankData database is created (if it doesn't exist) and the necessary tables are created.
During Part Two, you will create a server in WebSphere Application Studio Developer that is a Test Server running within the workbench environment.

In Part Three, you install the IBM Unit Test UDDI Registry in the workbench. You will use Cloudscape as the backend database for the Registry.

Exercise Instructions

Part One: Lab Setup

__1. Restore the original WebSphere configuration files.

__ a. At a Command Prompt, **CD \WebSphere\AppServer\bin**.

__ b. Type **restoreConfig.bat baseWASnode.zip**. This will recover the contents of several directories and configuration files from the **baseWASnode.zip** file, which was created as part of the Installation lab.

__2. The MyBank application (the application used throughout this lab) uses a database called BankData. This database needs to be created. At a Command Prompt, **CD \LabFiles50\WebServices\Bankdata**. Then type **BankData.bat wsdemo wsdemo1 Table.ddl**.

> ***NOTE:*** *There is a dependency in the loadBankDataBase.bat batch file that the BankData.bat batch file calls. If you do not have DB2 installed at C:\SQLLIB, you should update the setupVARs.bat file according to the instructions specified in the User Requirement section.*

__ a. This batch file drops the ACCOUNT and CUSTOMER tables in a database called BankData, re-creates the ACCOUNT table, and populates the ACCOUNT table with a few records.

__ b. After the BankData batch file completes, you should see the following message:

```
BankData database has been created and loaded.
```

Part Two: Create a WebSphere Test Server and Server Configuration with Application Developer

__1. Start WebSphere Studio Application Developer.

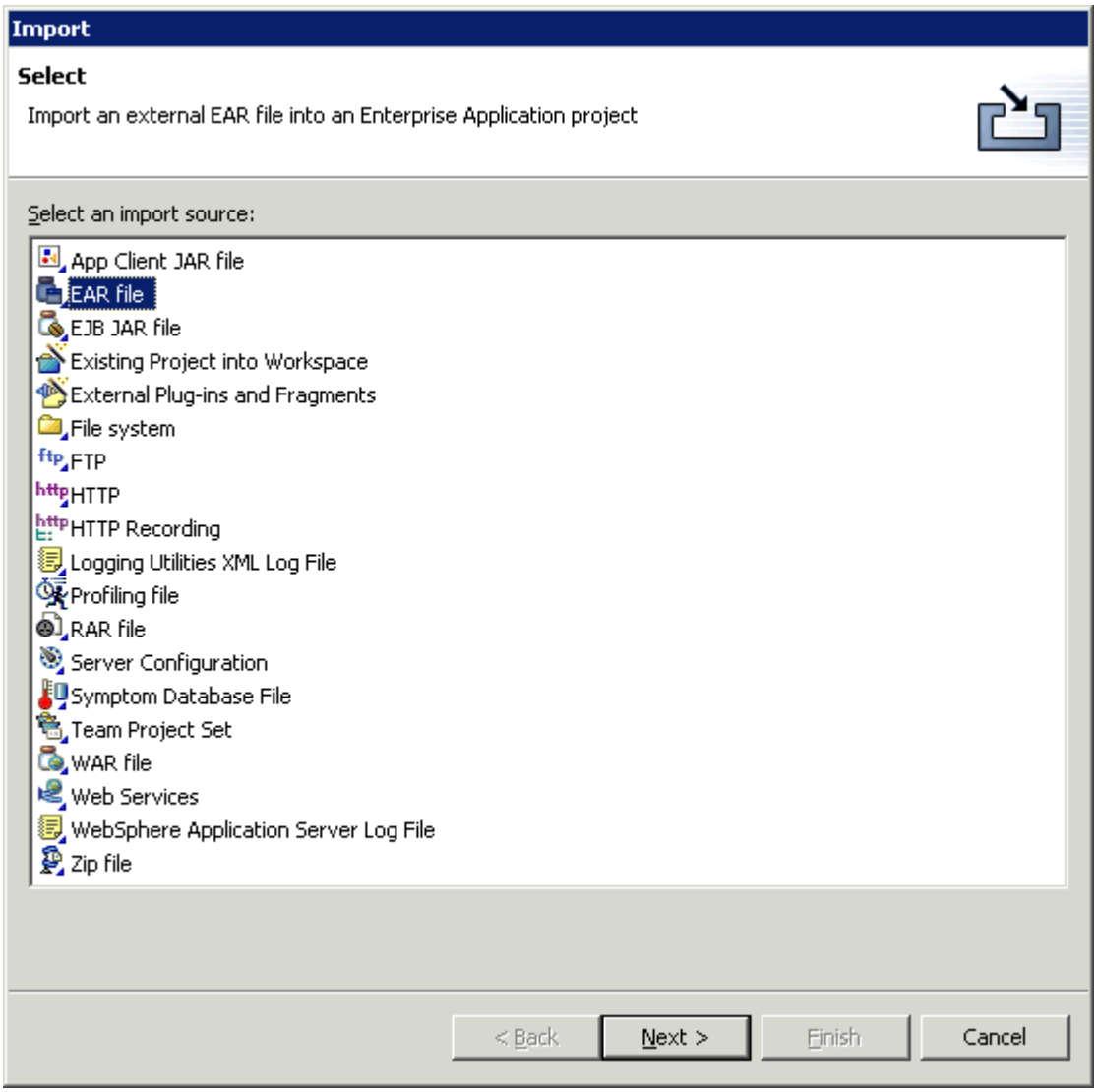
__ a. Select the menu **Start** > **Programs** > **IBM WebSphere Studio** and select **Application Developer 5.0**.

__ b. A dialog box will be displayed allowing you to select the location where you would like the workspace directory to be stored. Enter **C:\LabFiles50\WebServices\workspace** for the location and select **OK**. Application Developer will start with an empty workspace. An empty workspace will leave your existing workspace untouched and help avoid name conflicts between what you may already have in your workspace and what you will be creating in this lab.

__2. When WebSphere Application Studio Developer becomes active, you will need to switch to a J2EE perspective. If the J2EE perspective is not already opened, select **Window > Open Perspective > Other** and select **J2EE**. Click **OK**.

__3. Import the MyBank EAR file.

__ a. Select **File > Import... > EAR File**.

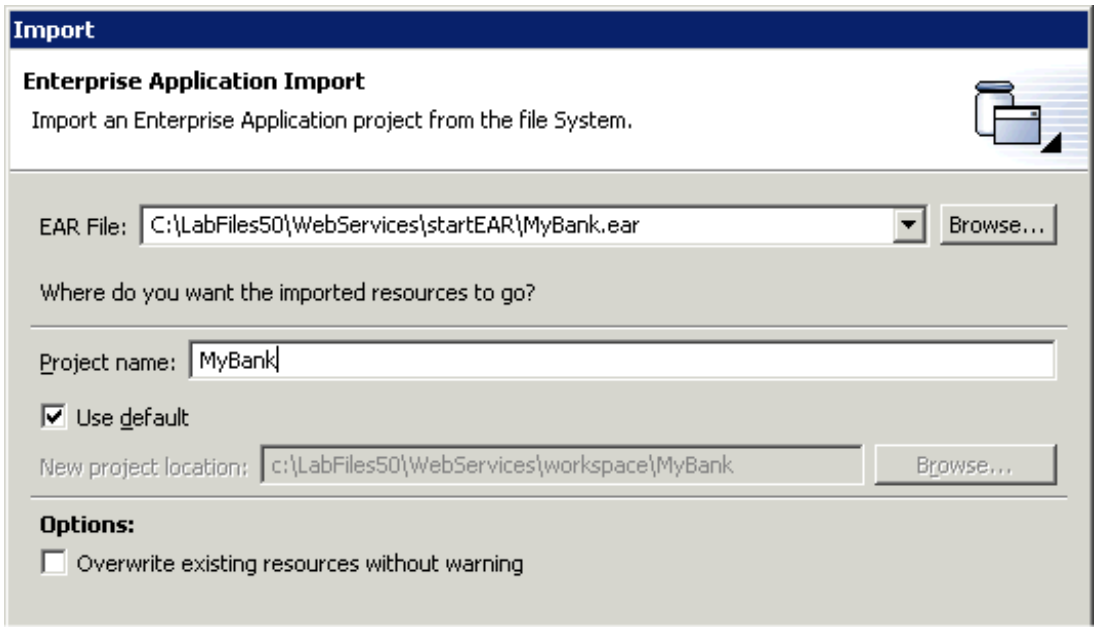


__ b. Click **Next**.

__ c. Click **Browse...** to the right of EAR file.

__ d. Navigate to **\LabFiles50\WebServices\startEAR**. Select **MyBank.ear** and click **Open**.

__ e. Type **MyBank** for the Project Name.

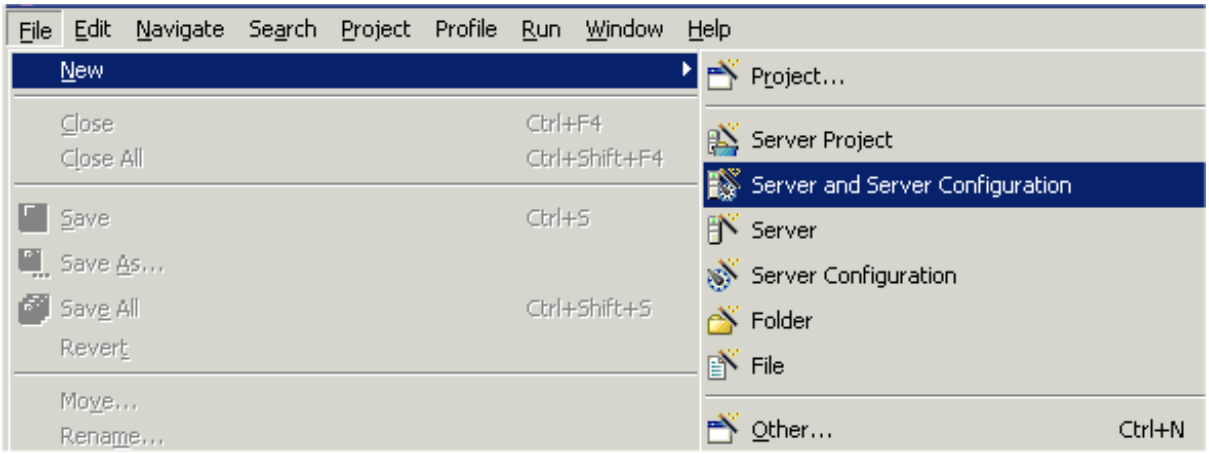


__ f. Click **Finish**.

__4. Create a WebSphere Test Environment Server.

__ a. First, you will configure a test server to test our Web Services within WebSphere Studio Application Developer. Select **Window > Open Perspective > Other...** and select **Server**. Click **OK**.

__ b. Next select **File > New > Server and Server Configuration**.



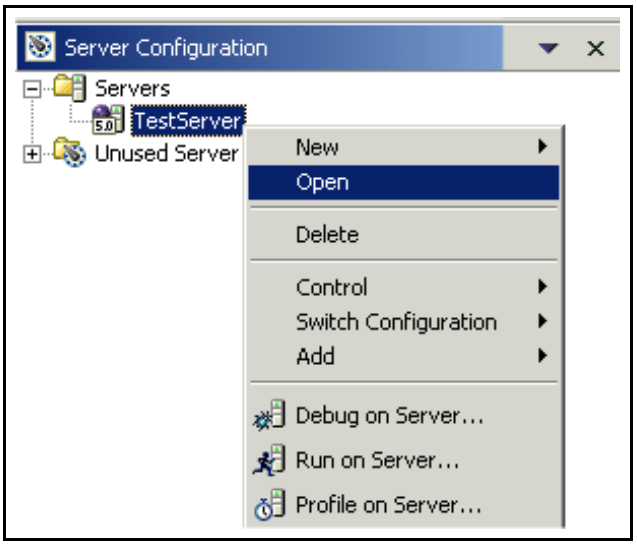
__ c. On the Create a new server and server configuration panel, type **TestServer** for the Server name and **TestServer** for the Folder.

__ d. For the Server type, expand **WebSphere version 5.0** and select **Test Environment**.

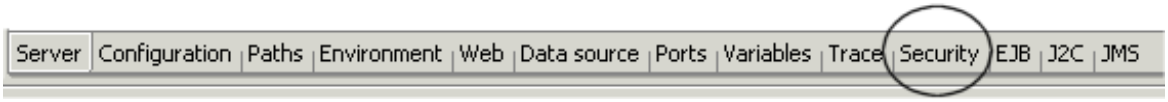
__ e. Click **Finish**.

__5. Create a Data Source for the Test Environment server.

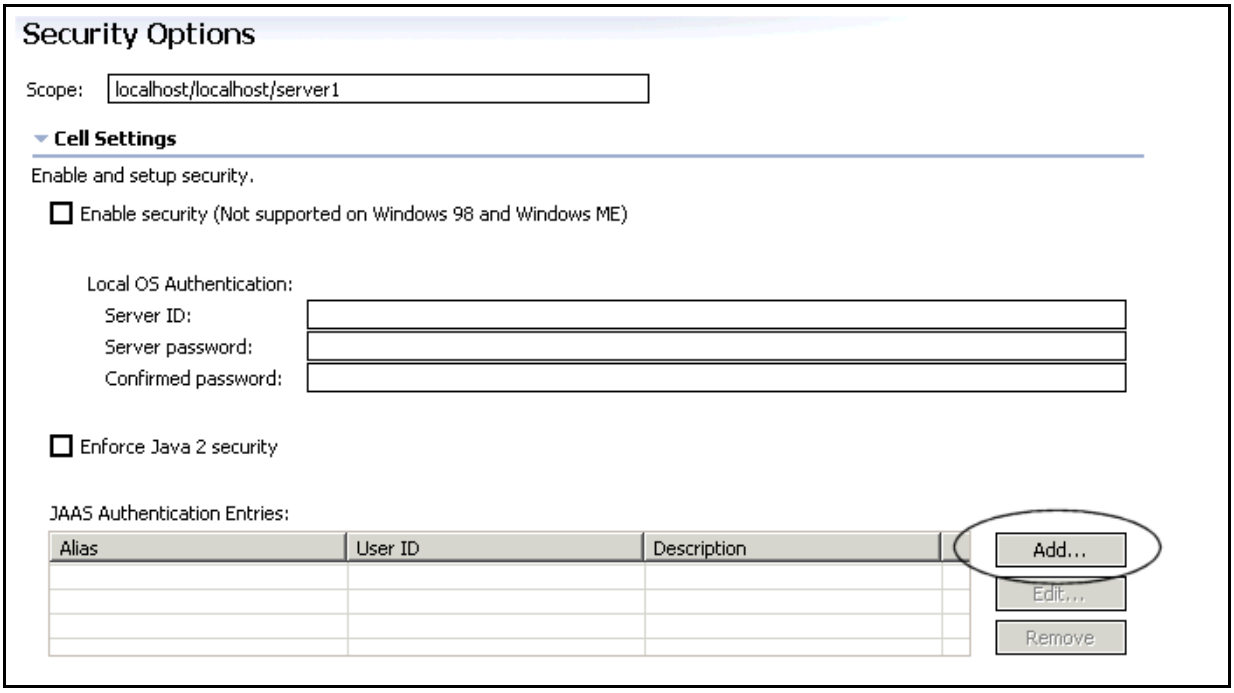
__ a. In the Server Configuration pane (lower left corner), expand **Servers**. Right-click **TestServer** and select **Open**.



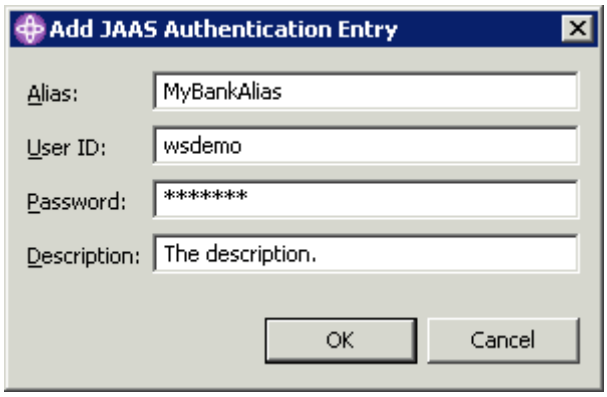
__ b. Select the **Security** tab as shown Below.



__ c. Click the **Add** button to the right of JAAS Authorization Entries.

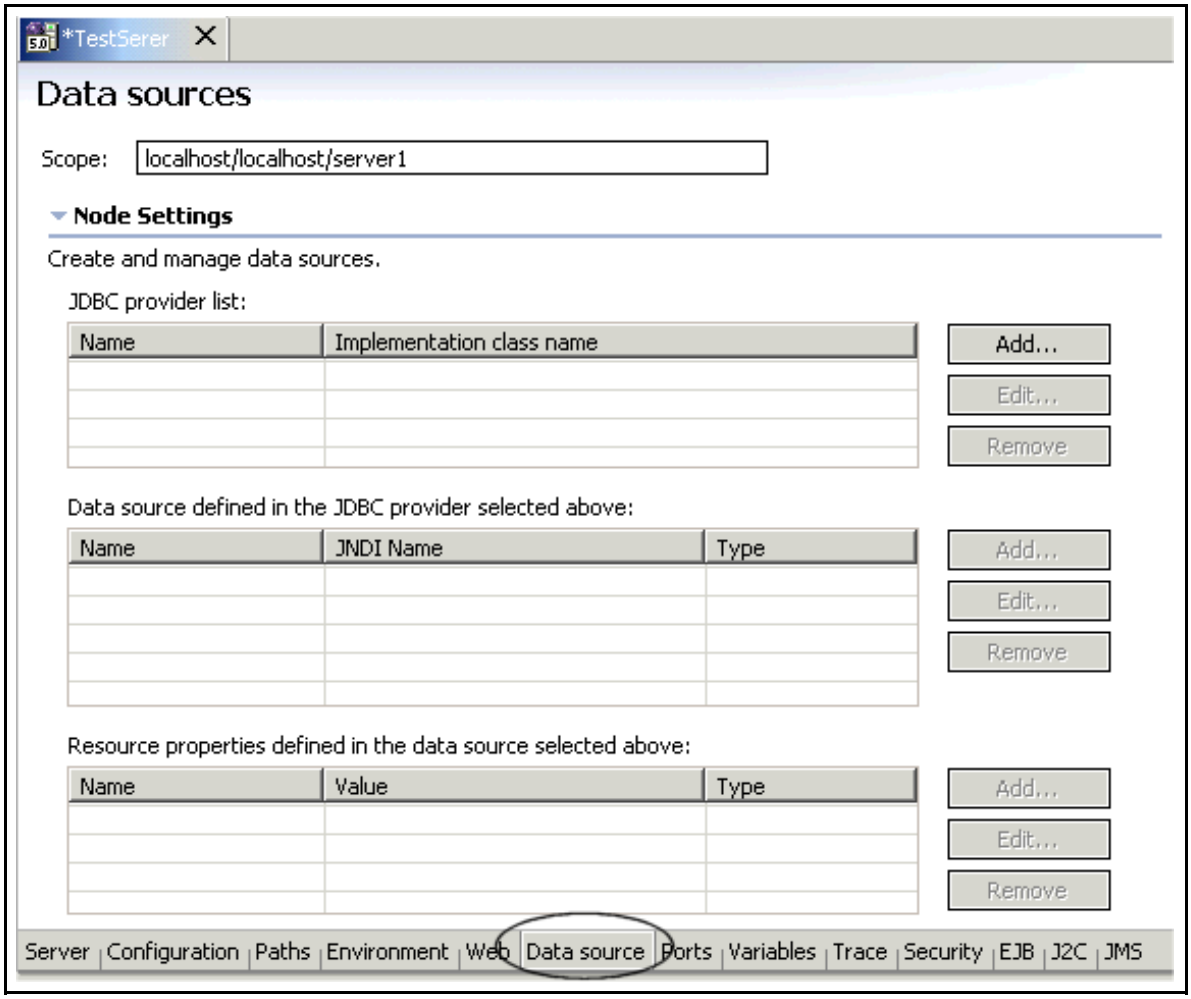


__ d. On the Add JAAS Authentication Entry panel, type **MyBankAlias** for the Alias name, type **wsdemo** for the Userid, type **wsdemo1** for the Password, and any text for the description.



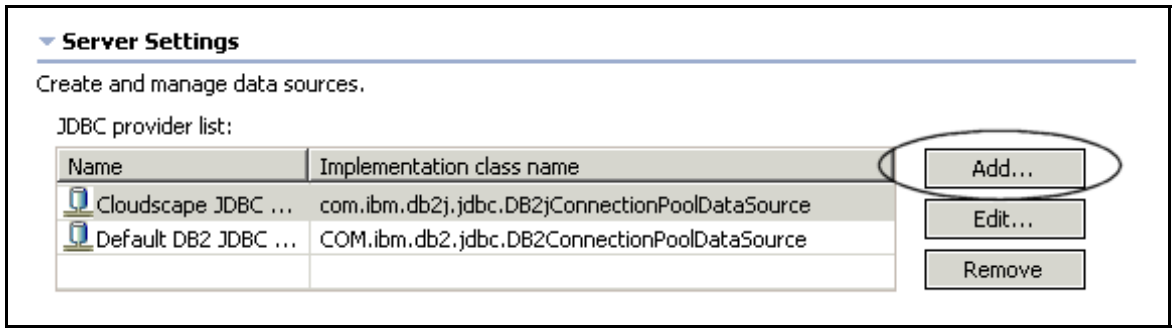
__ e. Click **OK**.

__ f. Select the **Data source** tab as shown below.

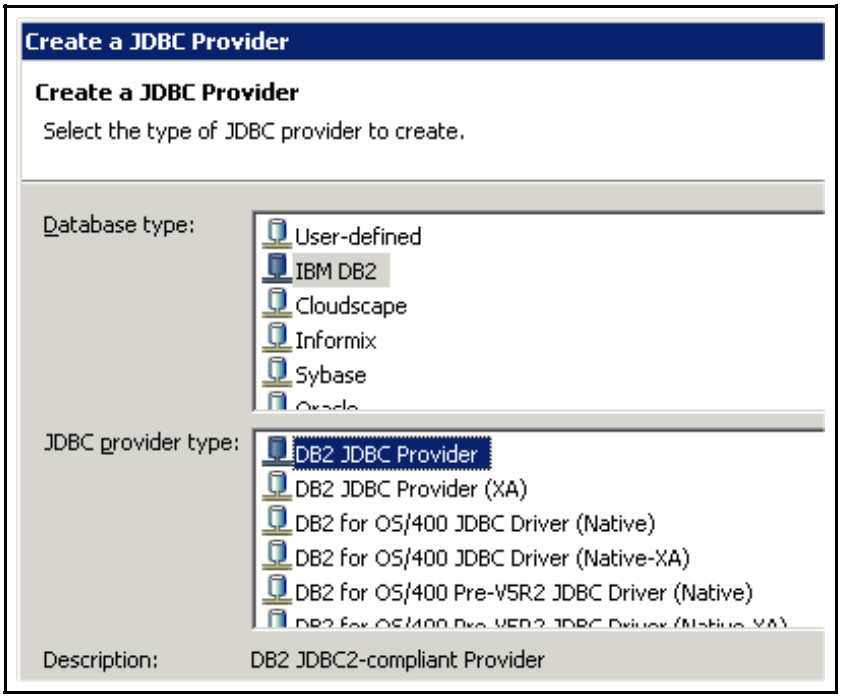


__ g. Scroll down until you see **Server Settings**.

__ h. Click the **Add** button to the right of JDBC provider list.

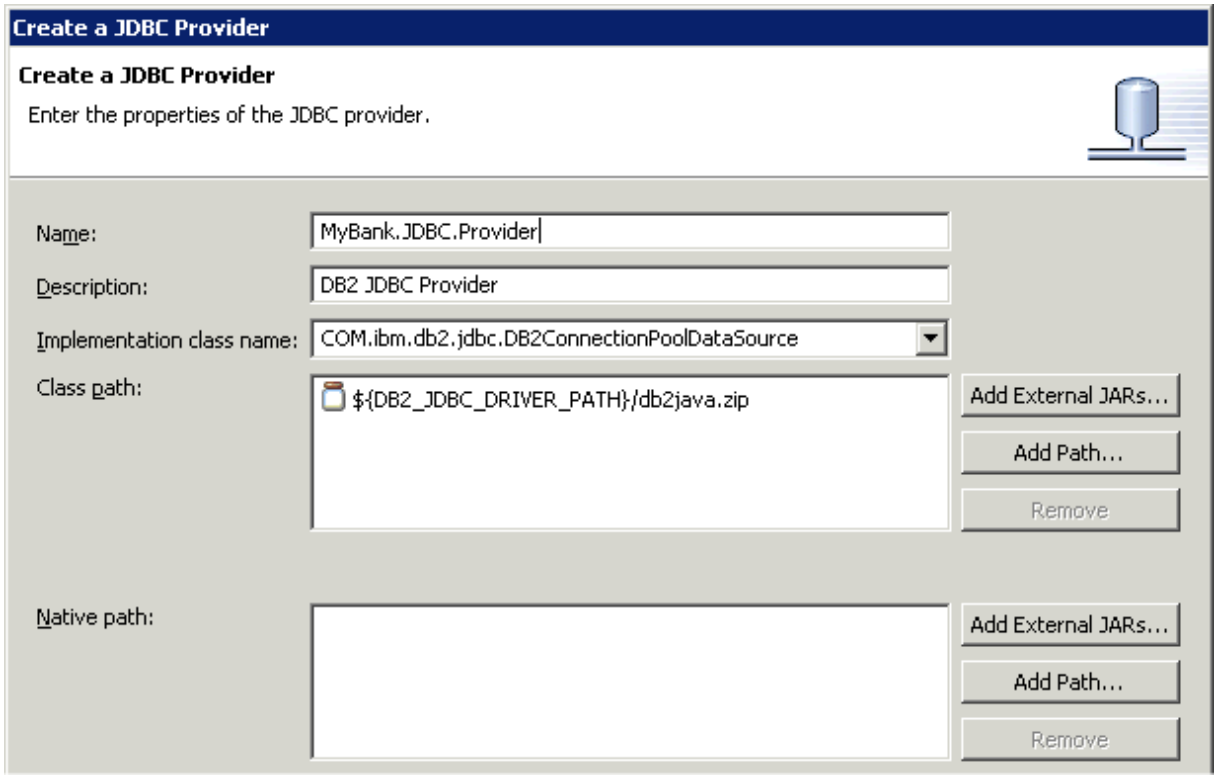


__ i. On the Create a JDBC Provider panel, select **IBM DB2** for the Database type and **DB2 JDBC Provider** for the JDBC provider type.



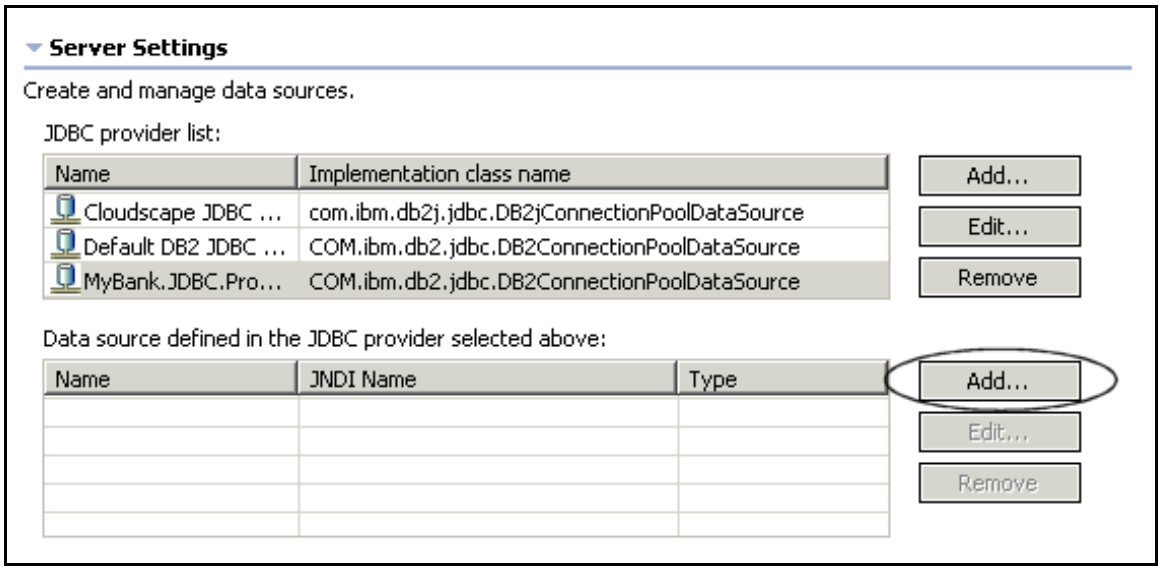
__ j. Click **Next.**

__ k. On the second panel (properties of the JDBC provider), type **MyBank.JDBC.Provider** for the Name. Accept the defaults for the remaining items.



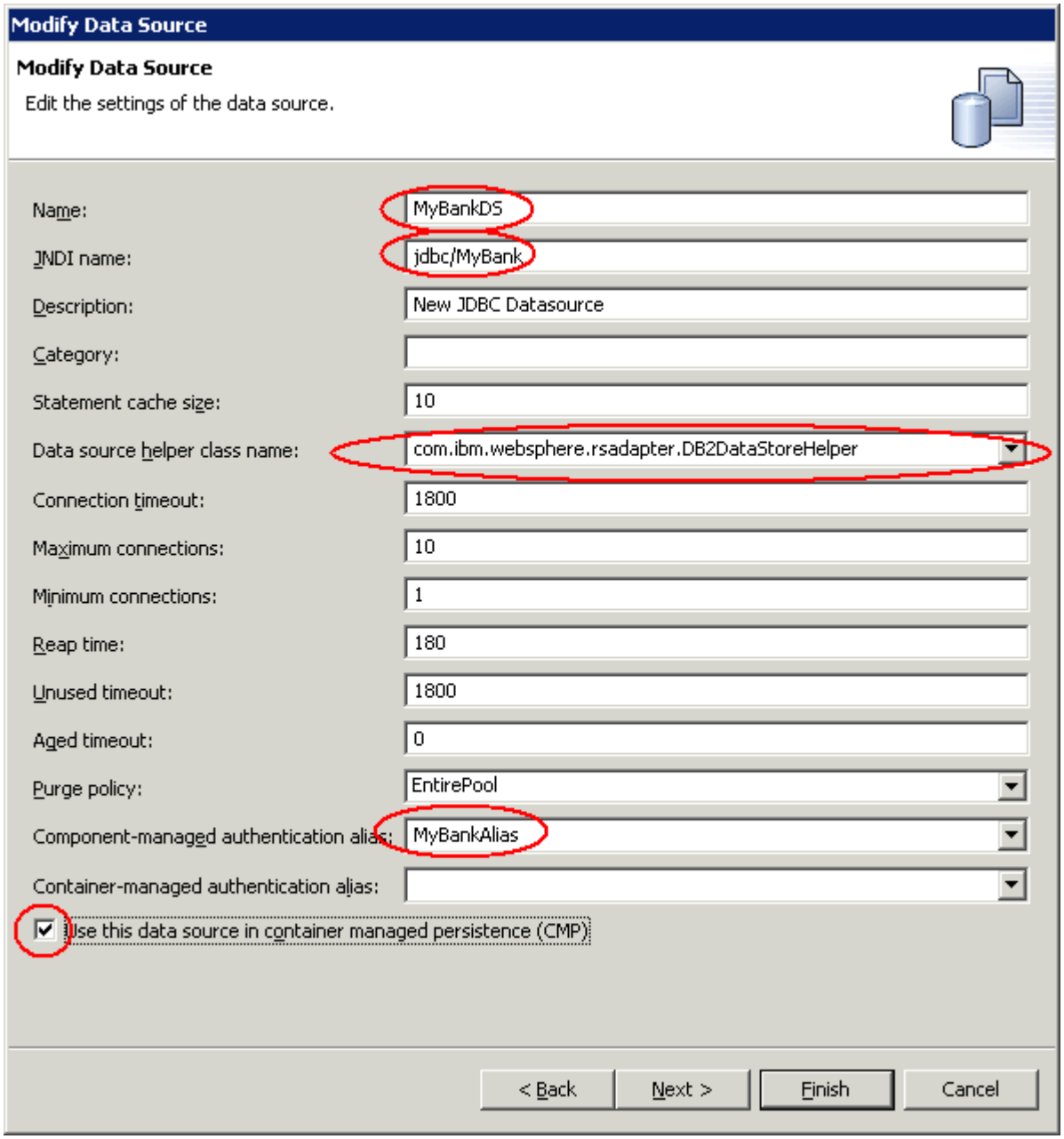
__ l. Click **Finish**.

__ m. Click **Add** to the right of the Data source defined in the JDBC provider selected above.



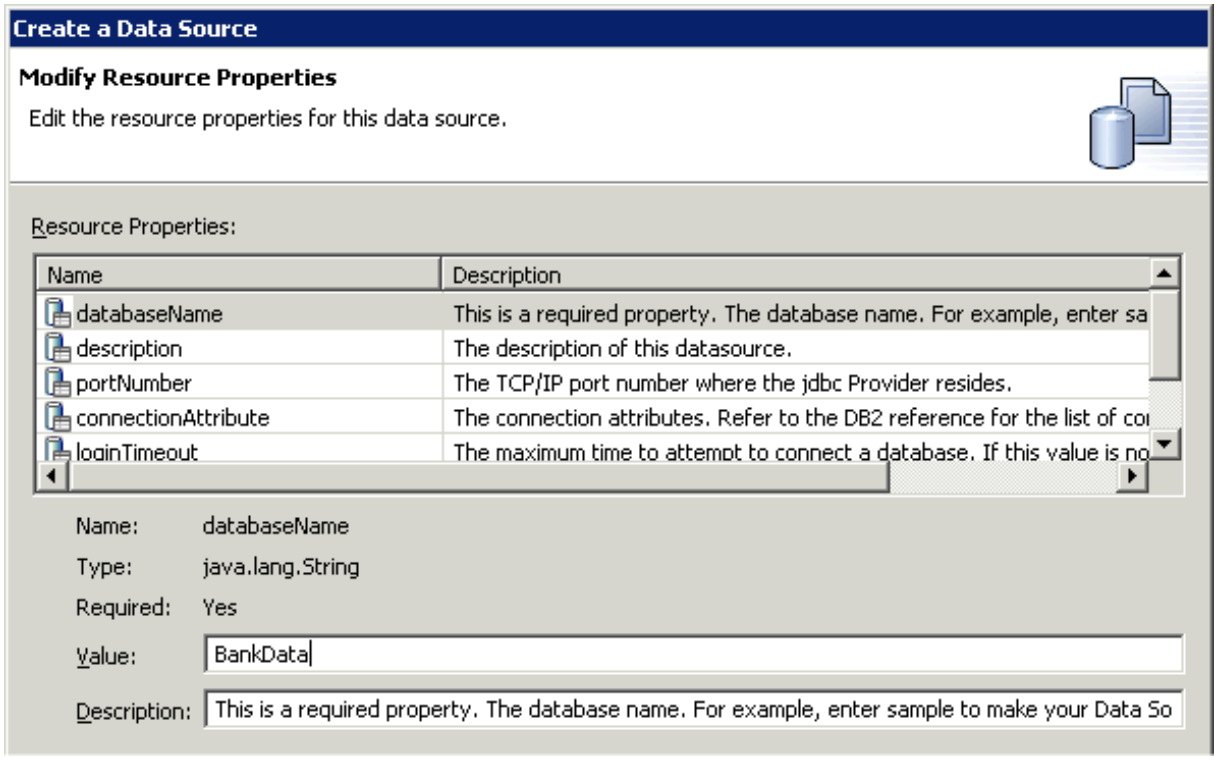
__ n. Select **Version 5.0 data source**. Click **Next**.

__ o. On the Modify Data Source panel, type **MyBankDS** for the Name. Type **jdbc/MyBank** for the JNDI Name. Ensure the Data source help class name is **com.ibm.websphere.rsadapter.DB2DataStoreHelper**. Select **MyBankAlias** for the Component-managed authentication alias. Ensure the checkbox to **Use this data source in container managed persistence (CMP)** is selected.



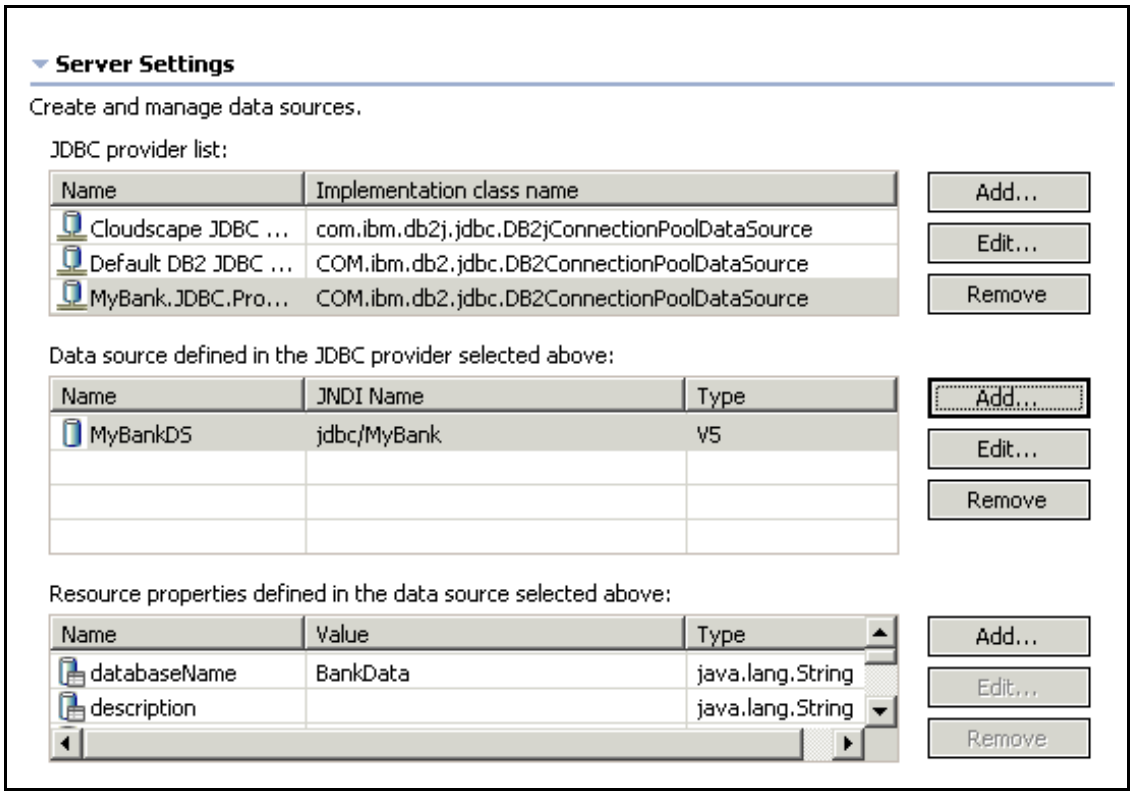
__ p. Click **Next**.

__ q. On the Modify Resource Properties panel, select databaseName and type **BankData** for the Value.



__ r. Click **Finish**.

__ s. The TestServer editor window should now look similar to the following graphic.

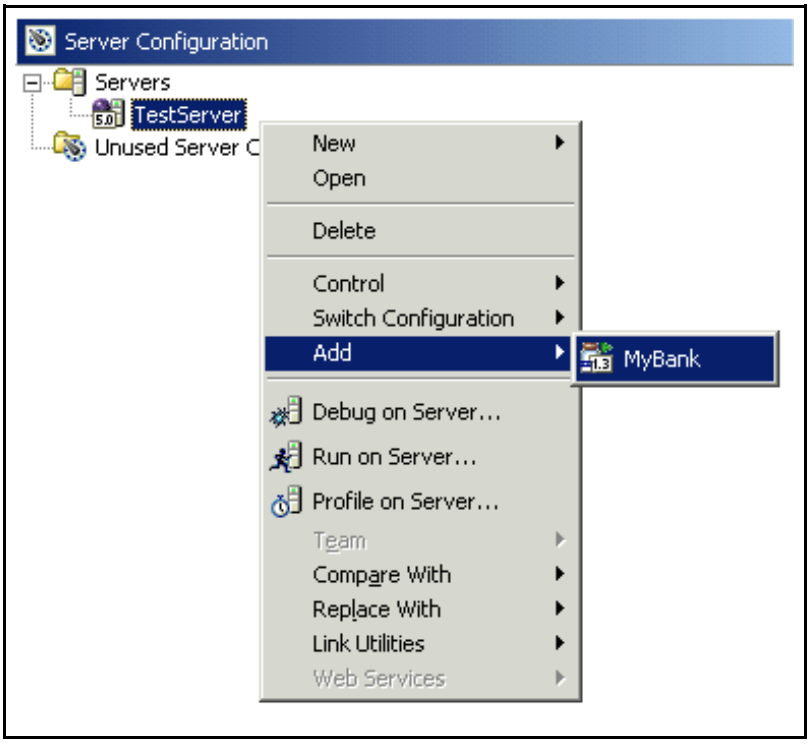


__ t. Save the Server Configuration, by typing **Ctrl+S**.

__ u. Next, **Close** the Server Configuration by clicking on the X on the right.

__6. Add the MyBank application to the Test Environment Server Configuration.

__ a. In the Server Configuration panel, **expand Servers**. Right-click TestServer and select **Add**, then choose **MyBank**.

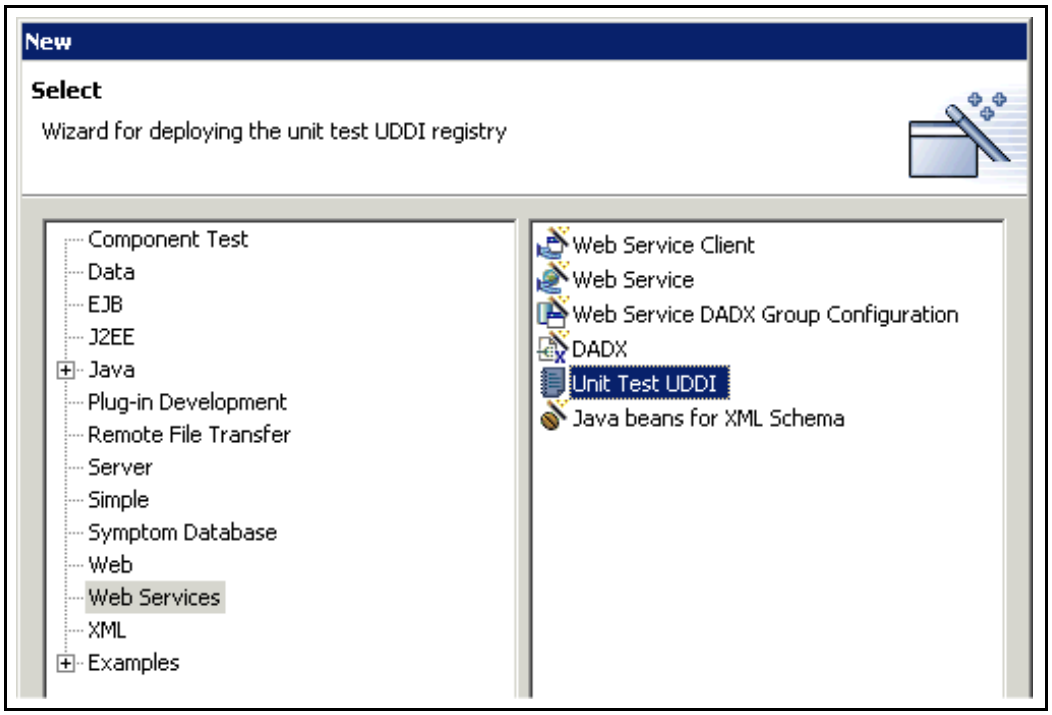


Part Three: Install the IBM Unit Test UDDI Registry within Application Developer

__1. Using a wizard, you can easily install the IBM Unit Test UDDI Registry.

__ a. Select the **J2EE Perspective**.

__ b. Select **File > New > Other...**. Then select the **Web Services** wizard in the left panel and **Unit Test UDDI** in the right Panel.

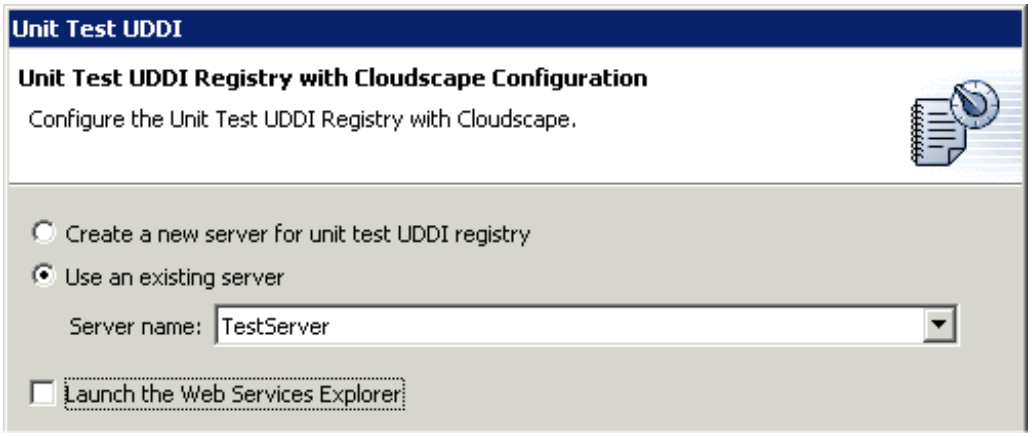


__ c. Click **Next**.

__ d. On the Unit Test UDDI Registry Configuration panel, ensure the **Private UDDI Registry For WAS v5 with Cloudscape** is selected for the Private UDDI Registry Type.

__ e. Click **Next**.

__ f. Select **Use an existing server** and ensure the Server name reads **TestServer**. **Unselect** Launch the Web Services Explorer.

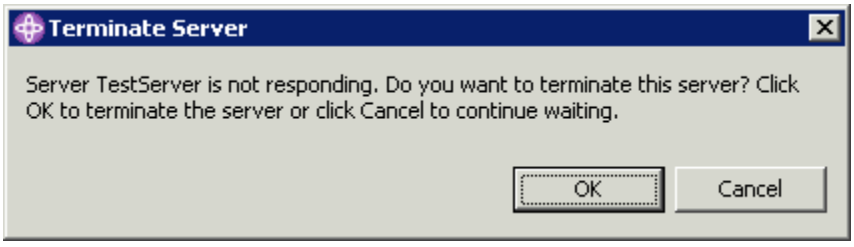


__ g. Click **Finish**.

__ h. If prompted about Modified Context Roots, select **Yes**.

The Unit Test UDDI Registry will be installed and deployed -- it is a J2EE application. This entire operation will take some time. The UDDI database is being created and loaded in Cloudscape.

__2. The IBM Unit Test UDDI Registry installation launches the WebSphere Test Server as part of its installation. This server should be stopped. Depending on how quickly you move from the previous step to this step, the TestServer may still be coming up. Regardless, you can stop the server. You *may* see a message like the following graphic, select OK to terminate.



__ a. Switch to the **Server Perspective**. Select **TestServer** from the **Server tab**. Right click on **TestServer** and select **Stop**. This will stop the WebSphere Test Environment server.

What you did in this exercise

In this lab exercise, you configured a WebSphere Test Server within the WebSphere Studio Application Developer and configured a Data Source to interact with a DB2 database. You also installed the Unit Test UDDI Registry in the workbench environment using Cloudscape as the backend database. There was no configuration of Cloudscape necessary.