IBM WEBSPHERE 5.0 Skills Transfer - LAB EXERCISE

Developing Web Services - Part 3

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.

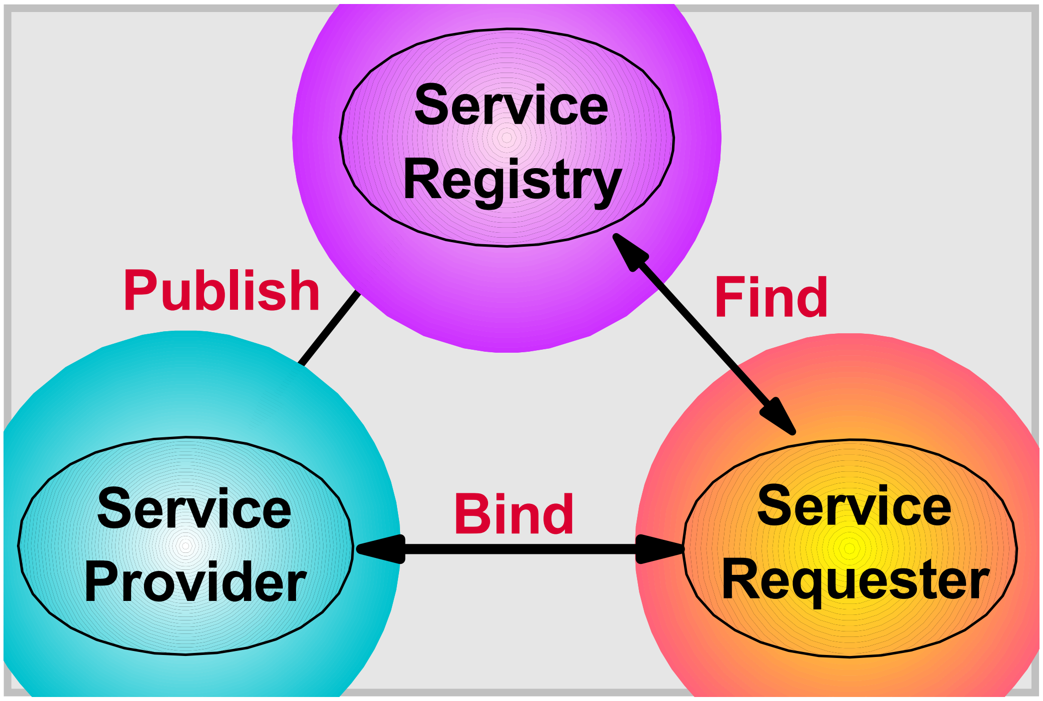
What You Should Be Able to Do

In this lab the focus is on working with the tools for creating and working with Web Services. By acting as the Service Provider added a Session EJB bean to the MyBank application which communicates with the Account EJB. You will also be programmatically publishing both the CreateAccount and TransferFunds Web Services to the Unit Test UDDI Registry whose backend database is DB2. Using the Web Services explorer from within WebSphere Application Studio Developer, you will find the TransferFunds service and write a sample application that will invoke the Web Service.

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, processing 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, the focus switches to the WebSphere Application Server Runtime environment. You begin by installing the UDDI Registry. The UDDI Registry is packaged with the WebSphere Network Deployment product.

In Part Two, you will install the MyBank J2EE application in the runtime environment. The installation and the creation of the JDBC provider, Data source, etc. have been automated with scripts.

During Part Three, you will programmatically publish both the CreateAccount and TransferFunds Web Services to the UDDI Registry via a Java program. In addition, using WebSphere Studio Application Developer, you will create another sample application.

In Part Four, the last part of the lab exercise, you will remove the MyBank Enterprise Application from the WebSphere Application Server runtime environment.

**** NOTE ** Solution instructions are at the end. The solution is provided in the case you do not want to do the exercise, or are unable to complete the lab and want to see the final solution. If you intend to go through the entire lab, start at Part One assuming you have met the requirements in the section titled "User Requirement" stated above.**

Exercise Instructions

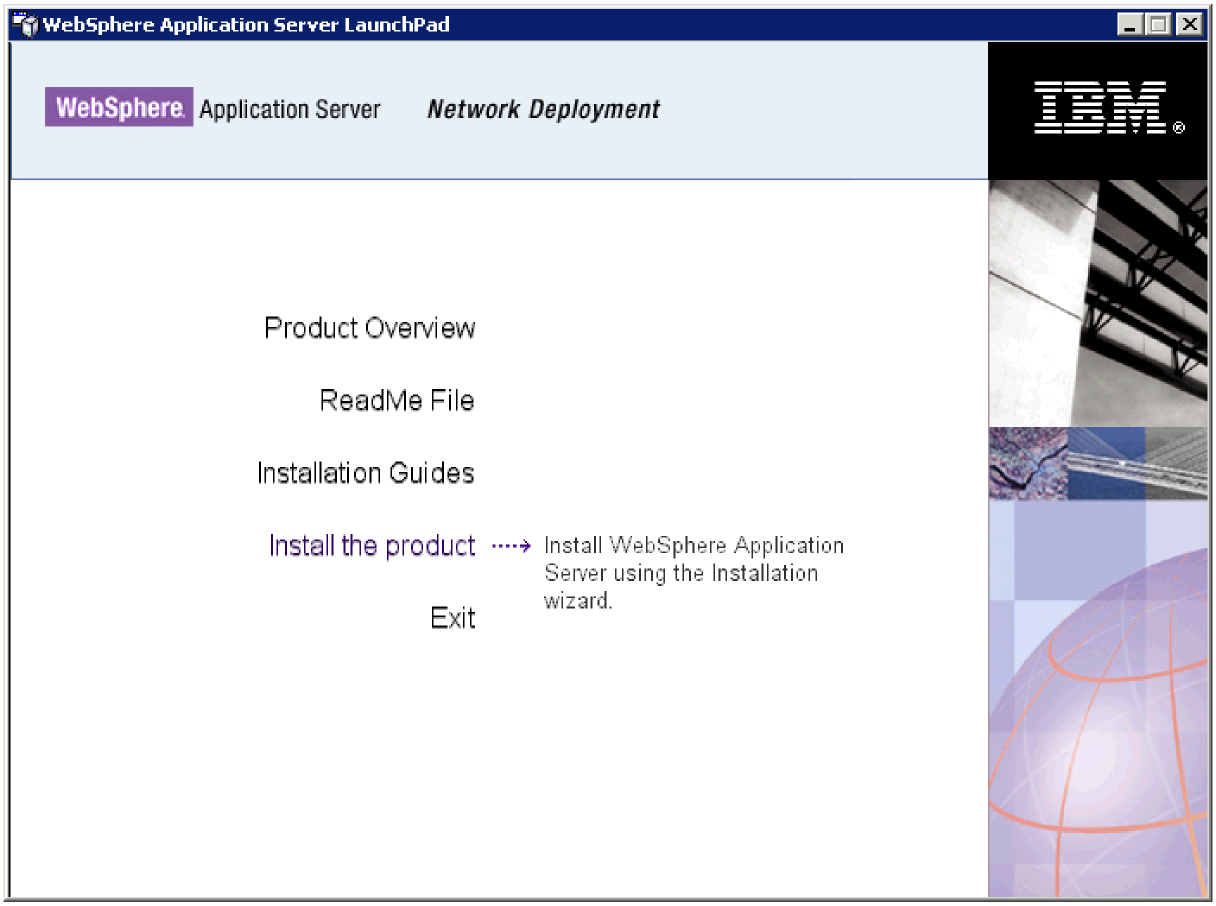
Part One: Install the UDDI Test Registry

__1. In a previous portion of the lab, you installed the Unit Test UDDI Registry with the WebSphere Studio Application Developer for testing purposes. To work with the UDDI Registry in a production mode, you will install the Registry via the WebSphere Application Server Network Deployment product package. NOTE: this part of the lab has several dependencies on the location of where the required software is installed. It is left up to the student to update paths if needed -- this can be done by updating the environment variables in \LabFiles50\setupVARs.bat as outlined in the User Requirement section.

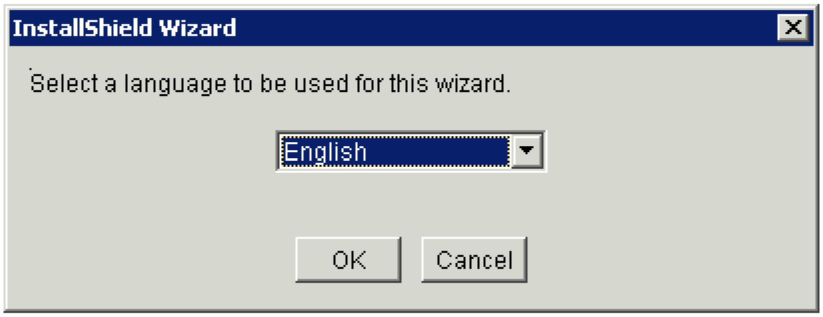
__2. At a Command Prompt, type **CD \Install\ND**, the location of the WebSphere Network Deployment package files . Run the install program by typing **LaunchPad.bat**.

__3. On the Language Selection panel that pops up, select your language. For example, **English**, and click **OK**.

__4. Select **Install the Product** to begin product installation.

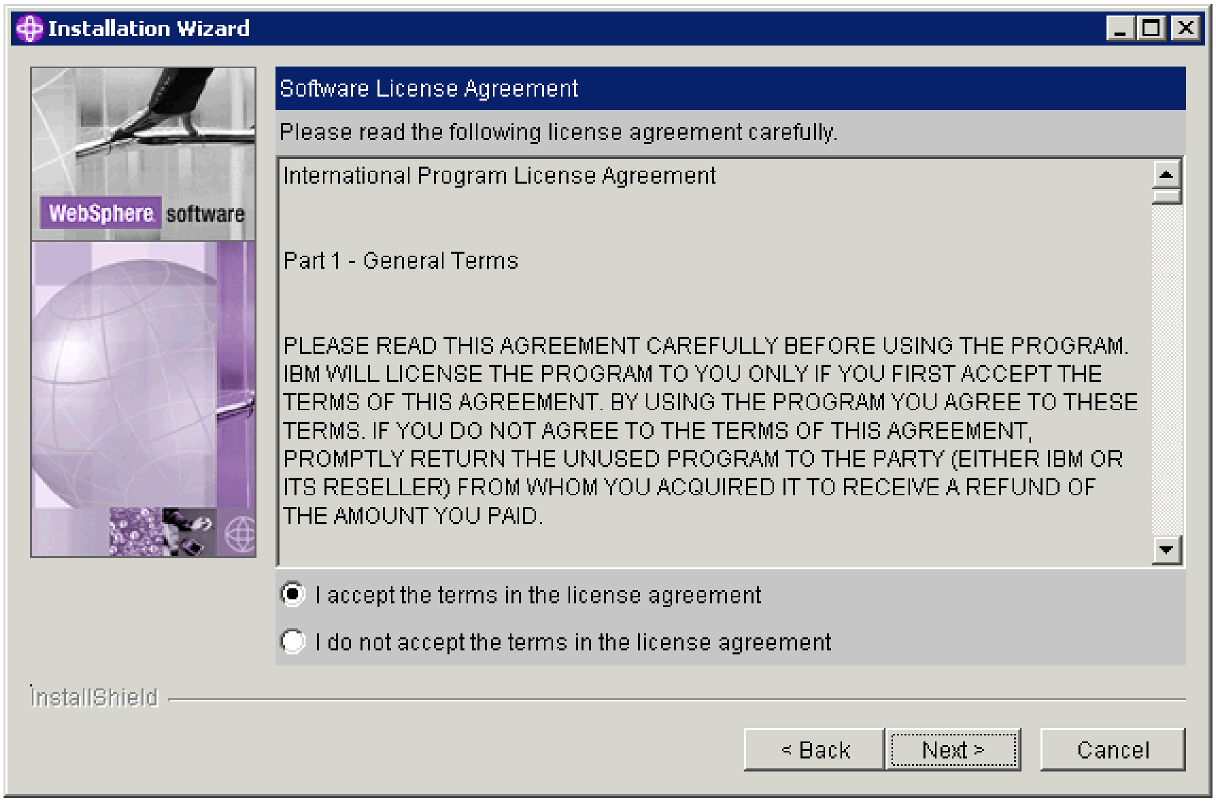

WebSphere Application Server LaunchPad
WebSphere Application Server
Network Deployment
IBM
Product Overview
ReadMe File
Installation Guides
Install the product
Install WebSphere Application Server using the Installation wizard.
Exit

__5. After selecting the Install the product option, you will receive another language selection panel. Select the same language as you selected in Step 3 above. For example, **English** and click **OK**.



__6. On the next Welcome panel, select **Next**.

__7. On the Software License Agreement panel, select **I accept the terms of the license agreement** and click **Next**.



Software License Agreement

Please read the following license agreement carefully.

International Program License Agreement

Part 1 - General Terms

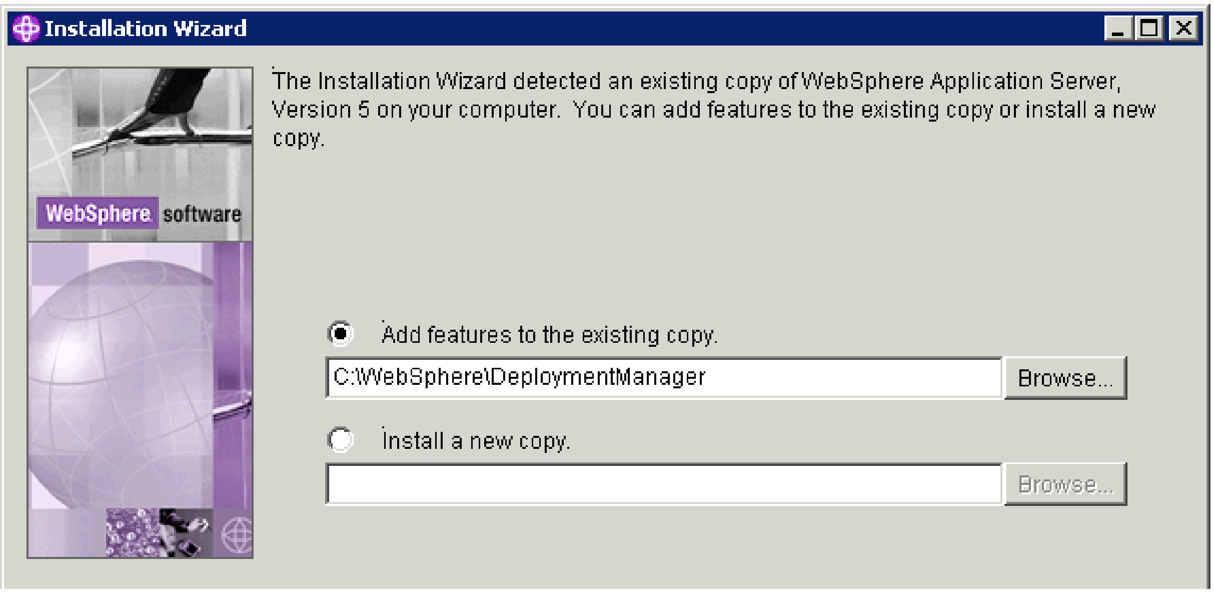
PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PROGRAM TO THE PARTY (EITHER IBM OR ITS RESELLER) FROM WHOM YOU ACQUIRED IT TO RECEIVE A REFUND OF THE AMOUNT YOU PAID.

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

__8. The WebSphere Installation wizard runs a software prerequisite check. If you don't have the necessary levels of the software components installed on your machine, you are warned, but the installation will continue. It is up to you to ensure you have the correct levels and fixpacks of the software components before running. For the purpose of this lab, you should not receive any warnings.

__9. Accept the defaults on the next panel if it is displayed. If this panel is not displayed, continue to the next step. Otherwise, click **Next**.



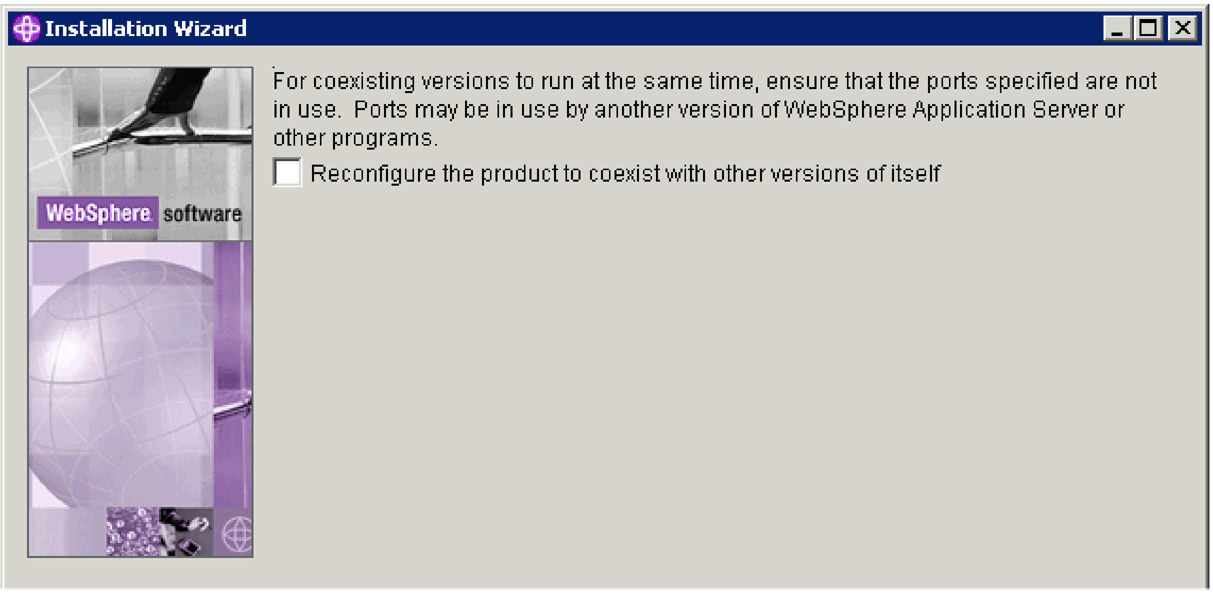
The Installation Wizard detected an existing copy of WebSphere Application Server, Version 5 on your computer. You can add features to the existing copy or install a new copy.

● Add features to the existing copy.

C:\WebSphere\DeploymentManager

○ Install a new copy.

__10. On the coexistence panel, do not select Reconfigure the product to coexist with other version of itself. Click **Next**.

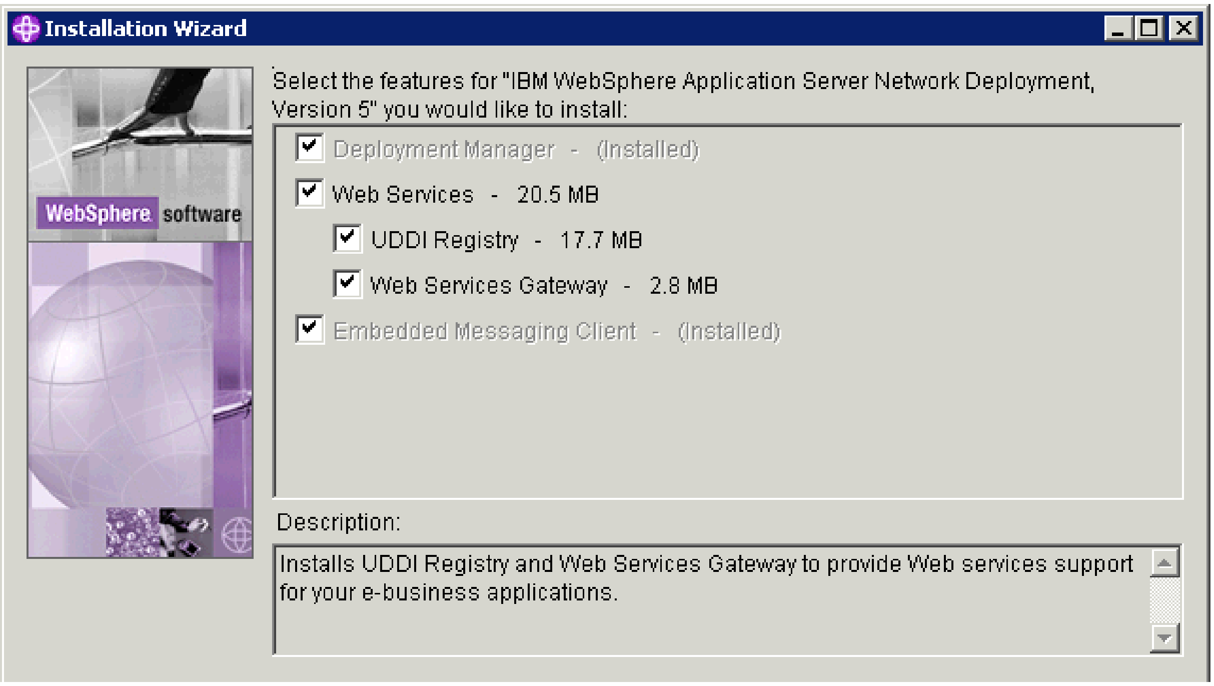


For coexisting versions to run at the same time, ensure that the ports specified are not in use. Ports may be in use by another version of WebSphere Application Server or other programs.

□ Reconfigure the product to coexist with other versions of itself

__11. On the next panel, ensure **Web Services, UDDI Registry**, and **Web Services Gateway** are selected. Click **Next**. Note: the panel you see may not have the Deployment

Manager and Embedded Messaging Client grayed out. Leave them selected.



Select the features for "IBM WebSphere Application Server Network Deployment, Version 5" you would like to install:

☑ Deployment Manager - (Installed)

☑ Web Services - 20.5 MB

☑ UDDI Registry - 17.7 MB

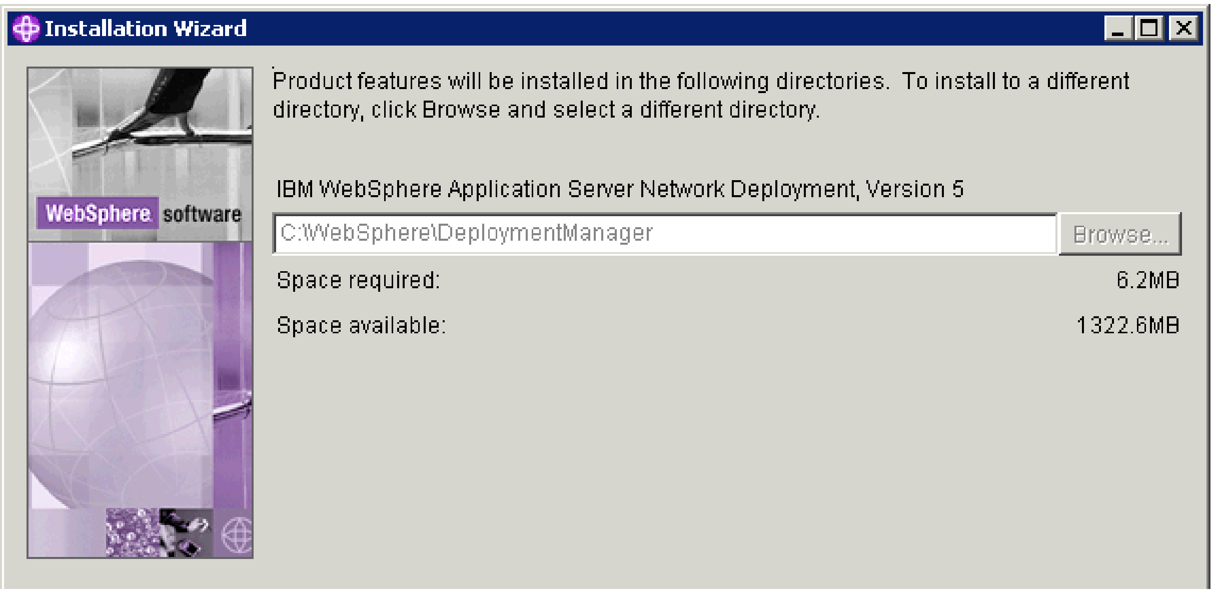
☑ Web Services Gateway - 2.8 MB

☑ Embedded Messaging Client - (Installed)

Description:

Installs UDDI Registry and Web Services Gateway to provide Web services support for your e-business applications.

__12. Ensure the installation path reads: **C:\WebSphere\DeploymentManager** (if you have the option of entering the path). Click **Next**.



Product features will be installed in the following directories. To install to a different directory, click Browse and select a different directory.

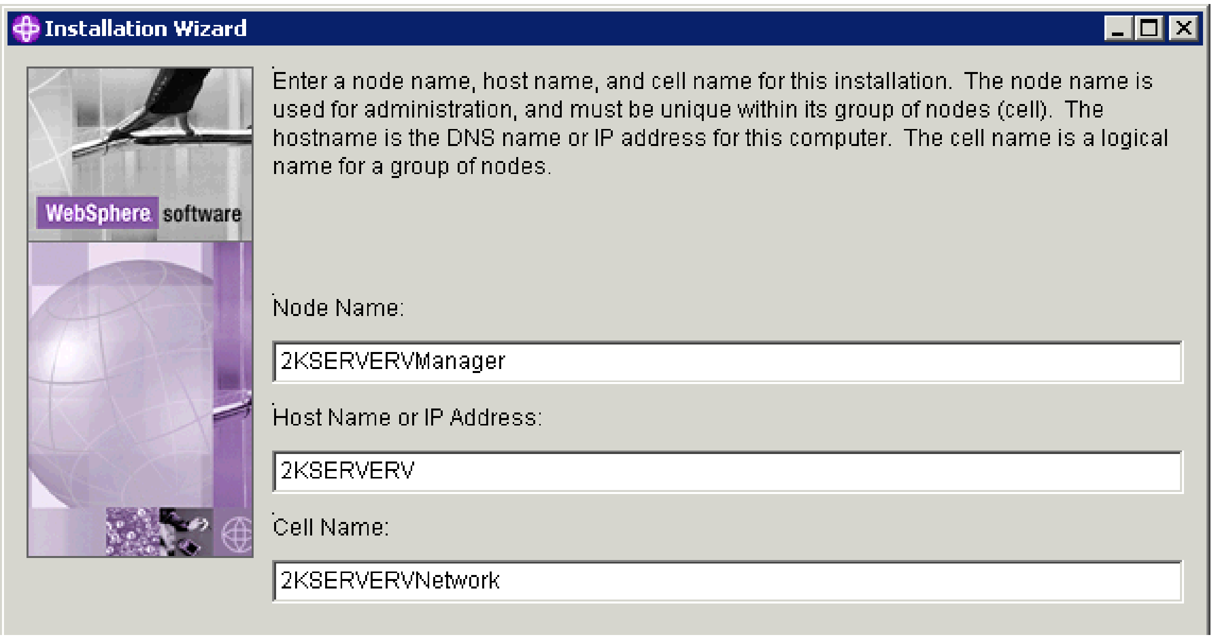
IBM WebSphere Application Server Network Deployment, Version 5

C:\WebSphere\DeploymentManager

Space required: 6.2MB

Space available: 1322.6MB

__13. Accept the defaults for the Node Name, Host Name or IP Address, and Cell Name. An example is below. Click **Next**.



Enter a node name, host name, and cell name for this installation. The node name is used for administration, and must be unique within its group of nodes (cell). The hostname is the DNS name or IP address for this computer. The cell name is a logical name for a group of nodes.

Node Name:

2KSERVERVManager

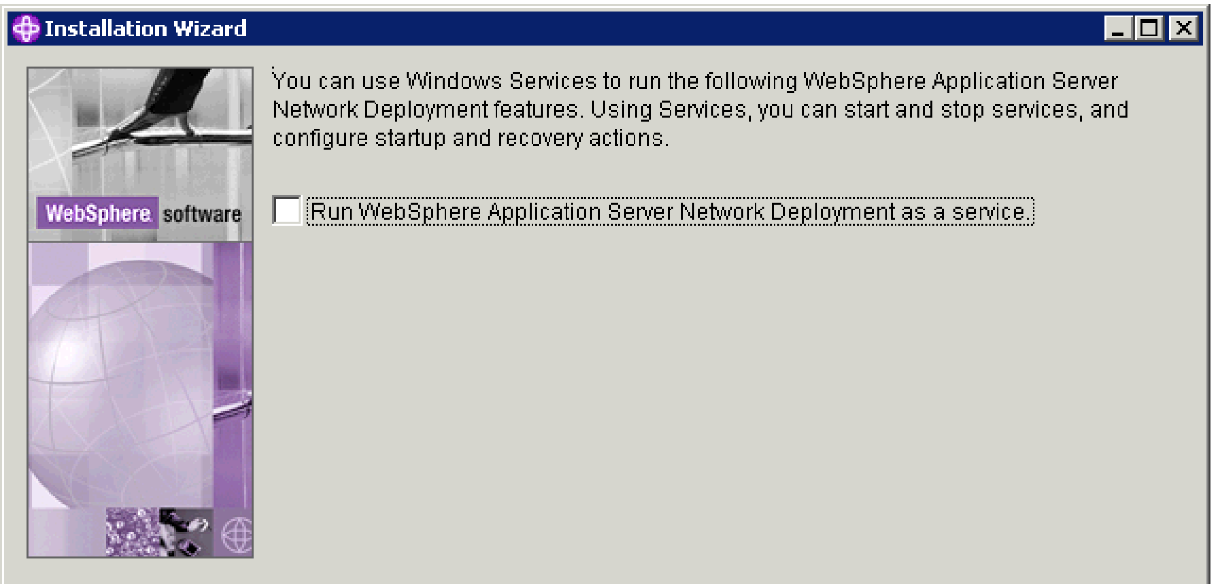
Host Name or IP Address:

2KSERVERV

Cell Name:

2KSERVERVNetwork

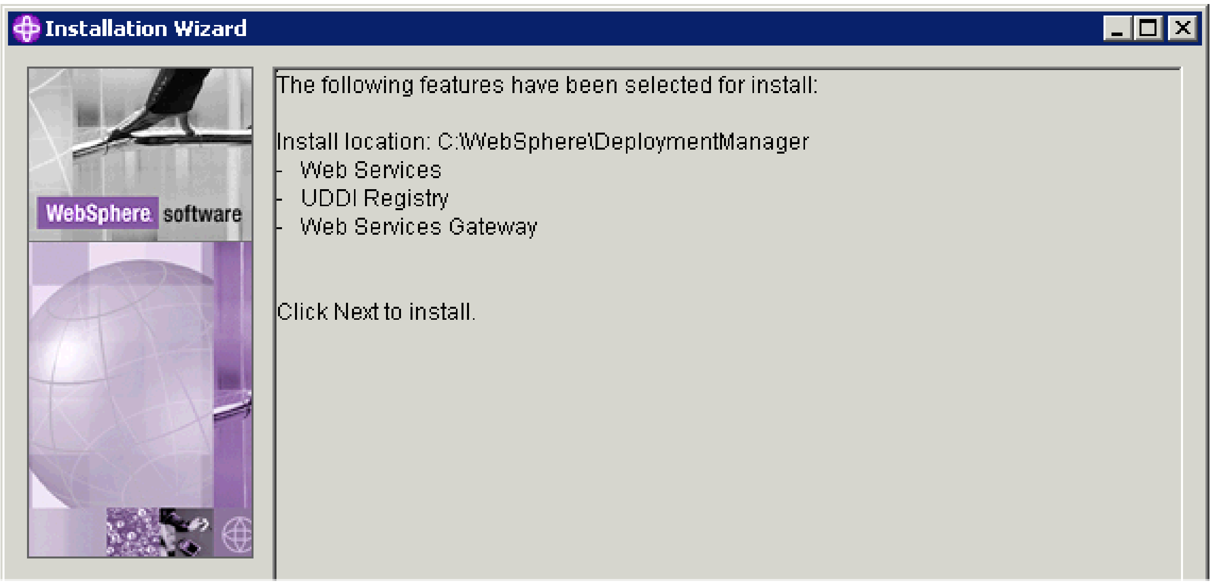
__14. You will not run Network Deployment as a service. **Un-select Run WebSphere Application Server Network Deployment as a service**. Click **Next**.



You can use Windows Services to run the following WebSphere Application Server Network Deployment features. Using Services, you can start and stop services, and configure startup and recovery actions.

☐ Run WebSphere Application Server Network Deployment as a service.

__15. Review your selections and click **Next**.



The following features have been selected for install:

Install location: C:\WebSphere\DeploymentManager
- Web Services
- UDDI Registry
- Web Services Gateway

Click Next to install.

__16. After the installation complete, you are prompted to Register the product. For our training purposes, you will not Register the product. Therefore, **un-select Register this product**. Click **Next**.

__17. On the **WebSphere Application Server - First Steps** window, select **Verify Installation**. This will run the install verification test to ensure successful installation.

__18. You should see a "IVT Verification Succeeded" message. Click **Exit**.

__19. Click **Finish**.

__20. Click **Exit** on the WebSphere Application Server Network Deployment LaunchPad window.

__21. Open a command prompt, and **CD \WebSphere\DeploymentManager\bin**. Type **stopmanager**.

__22. Install the WebSphere Application Server Network Deployment FixPak 1.

__ a. Before installing or uninstalling updates, stop all WebSphere Application Server processes, as well as related component processes such as the IBM HTTP Server and embedded messaging. To stop the service, right-click **My Computer**, select **Manage**, select **Services and Applications**, and select **Services**. Stop all services that begin with "**IBM HTTP**…". Click **Next**.

__ b. Navigate to the **C:\install\nd5.0.1** directory.

__ c. Double-click on the file **was50_nd_fp1_win.zip** (the .zip extension may be hidden).

__ d. Click **Extract** and enter **C:\WebSphere\DeploymentManager\update**.

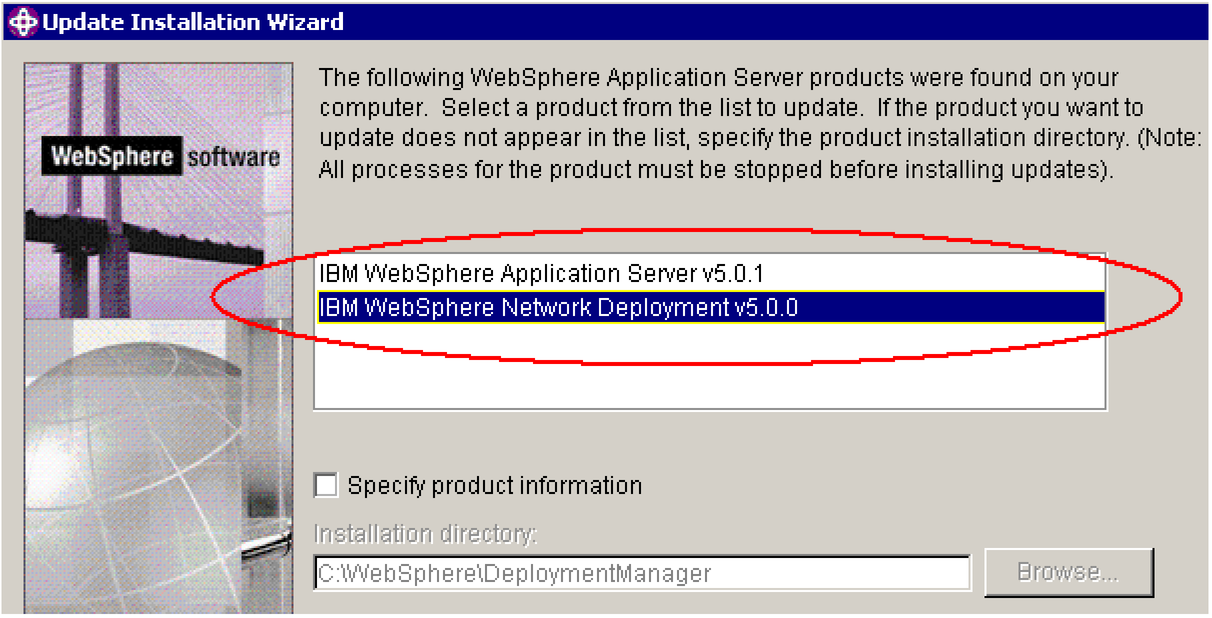
__ e. Click **Extract**.

__ f. Close WinZip.

__ g. At a Command Prompt, type **CD \WebSphere\DeploymentManager\update**. Execute the updateWizard batch file by typing **updateWizard.bat**.

__ h. Read the Welcome Page and stop any application server processes that are running. NOTE: This should be already done in the first step under "Install the WebSphere Application Server Network Deployment FixPak 1". Click **Next**.

__ i. Select **IBM WebSphere Network Deployment v5.0.0**. Click **Next**.



The following WebSphere Application Server products were found on your computer. Select a product from the list to update. If the product you want to update does not appear in the list, specify the product installation directory. (Note: All processes for the product must be stopped before installing updates).

IBM WebSphere Application Server v5.0.1
IBM WebSphere Network Deployment v5.0.0

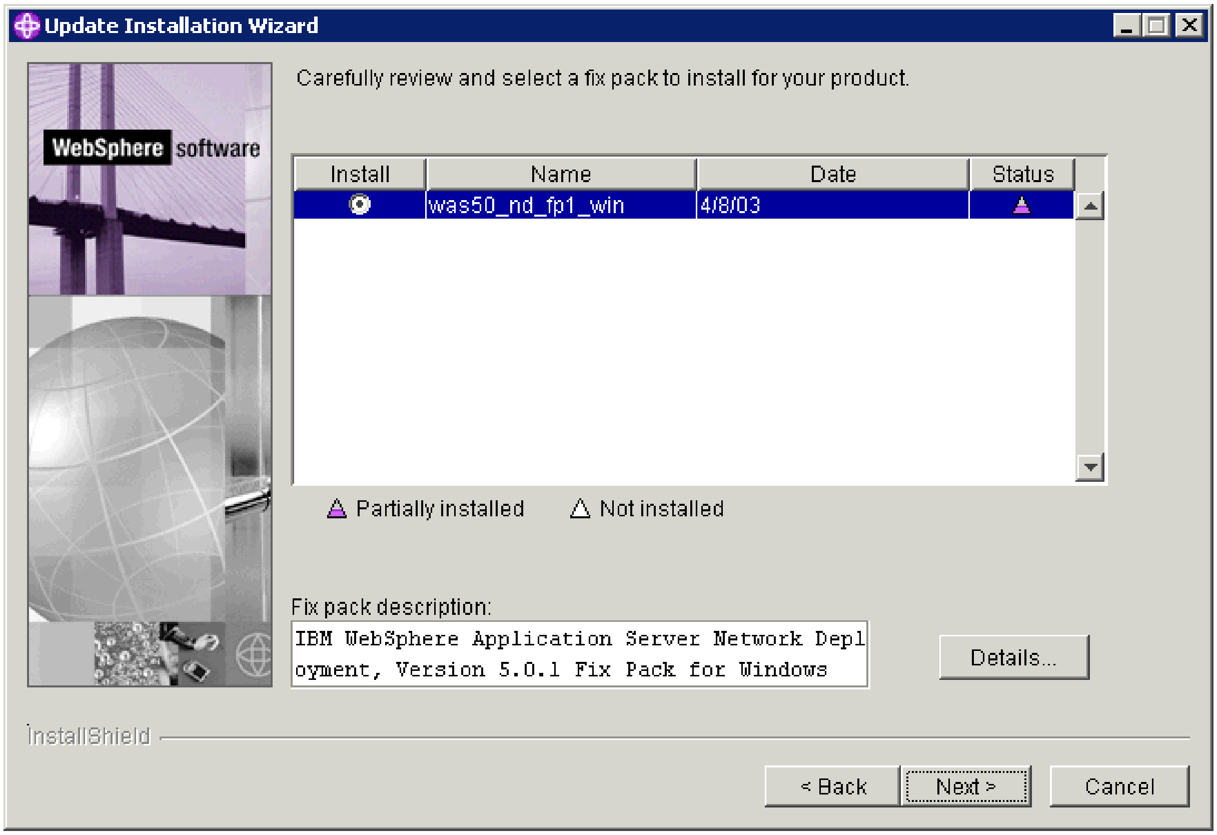
□ Specify product information

Installation directory:
C:\WebSphere\DeploymentManager

__ j. Ensure **Install fix packs** is selected. Click **Next**.

__ k. Accept the default Fix pack directory and click **Next**.

__ l. Verify that the **was50_nd_fp1_win** fix pack is selected.



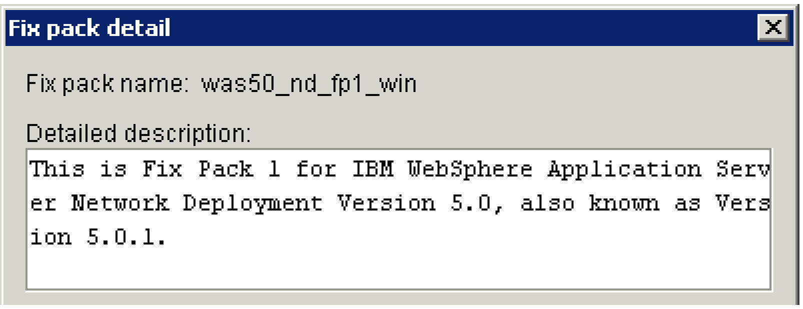
Carefully review and select a fix pack to install for your product.

Install	Name	Date	Status
●	was50_nd_fp1_win	4/8/03	▲

Fix pack description:

IBM WebSphere Application Server Network Deployment, Version 5.0.1 Fix Pack for Windows

__ m. Click on **Details** and verify that the fix pack is the right one. Then click **OK**.



Fix pack name: was50_nd_fp1_win

Detailed description:

This is Fix Pack 1 for IBM WebSphere Application Server Network Deployment Version 5.0, also known as Version 5.0.1.

__ n. Click **Next**.

__ o. Accept the default directories for the IBM HTTP Server and Embedded Messaging by clicking **Next**.

__ p. On the summary window, click **Next**.

__ q. After the install completes, click **Finish**.

__23. At this point, ensure the setupVARs.bat file in \Labfiles50\WebServices is up-to-date with where you have installed WebSphere Network Deployment

__24. At a Command Prompt, type **CD \WebSphere\AppServer\bin**. Start the WebSphere Application Server by typing **startserver server1**. After successful startup of the application server, you should see resulting console messages like the following:

```
C:\WebSphere\AppServer\bin>startserver server1
ADML0100I: Reading configuration for server: server1
ADML0200I: Server launched. Waiting for initialization status.
ADML0000I: Server initialization completed. Process id is: 2520
```

__25. At a Command Prompt, **CD \WebSphere\DeploymentManager\properties**.

__26. Type **notepad uddi.properties** to edit the properties file.

__27. Cloudscape is the normal default. Ensure DB2 is listed as the database persister. If DB2 is not the persister, modify the section in the uddi.properties (place a '#' sign in front of the Cloudscape line and remove the '#' sign from the DB2 line) to look like the following:

```
#######################################################
# Persister implementation.
#
# The persistence store for the UDDI Registry.
#
# Valid values are currently
# for DB2:
#   persister=DB2
# for Cloudscape:
#   persister=Cloudscape
#
# persister=string
############################################################
#persister=Cloudscape
persister=DB2
```

__28. Next, **save** and **close** the uddi.properties file.

__29. Install the DB2 UDDI database.

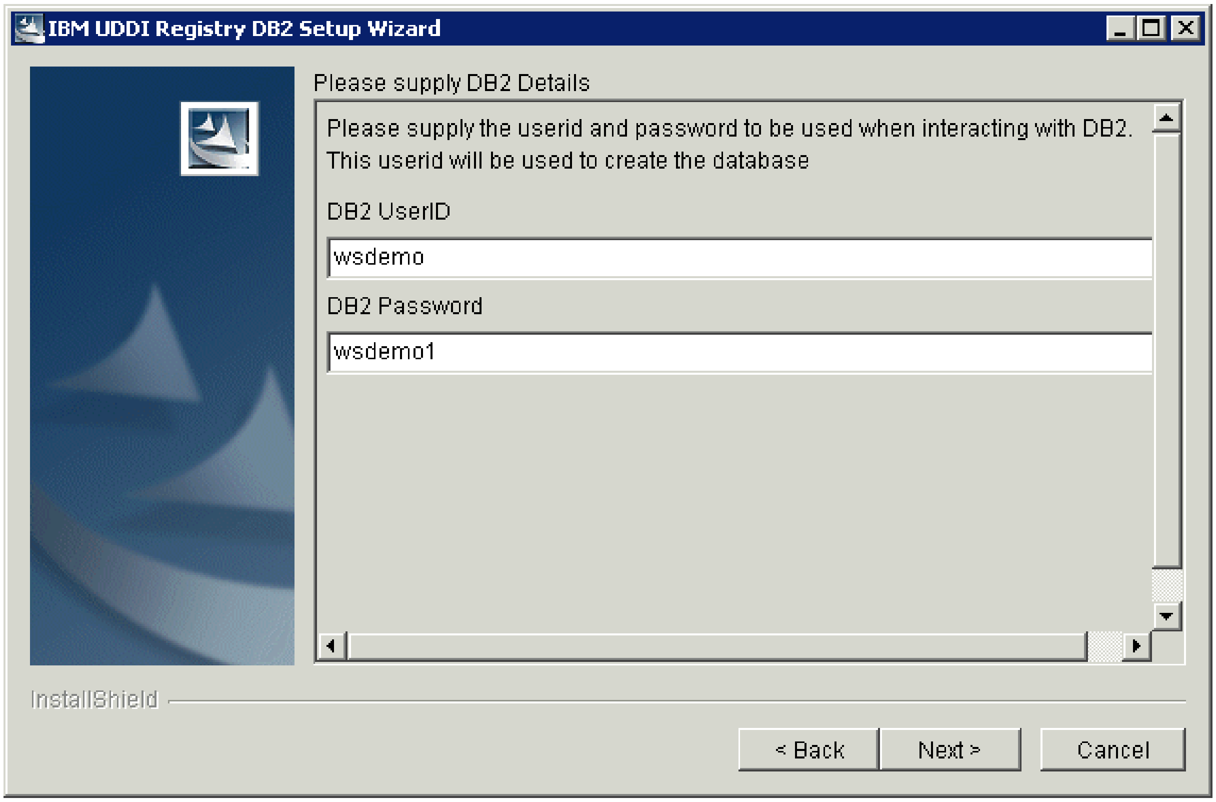
__ a. At a Command Prompt, **CD \Labfiles50\WebServices\UDDIRegistry**.

__ b. Type **SetupDB2.bat**. This will launch the UDDI DB2 setup wizard.

__ c. Select the language you wish to use. For example, English is the default. Click **OK**.

__ d. On the "IBM UDDI Registry DB2 Setup Wizard" panel, click **Next**.

__ e. Specify **wsdemo** and **wsdemo1** for the userid and password respectively.



Please supply DB2 Details

Please supply the userid and password to be used when interacting with DB2.
This userid will be used to create the database

DB2 UserID: wsdemo

DB2 Password: wsdemo1

__ f. Click **Next**.

__ g. After validating the userid and password, the UDDI database installation begins and you will see a message indicating the database creation is in progress.

__ h. The next panel that appears should indicate successful completion of the UDDI database creation. Click **Next**.

__ i. Click **Finish**.

__30. In the same Command Prompt window, **CD \Labfiles50\WebServices\UDDIRegistry**. Type **SetupUDDI.bat <db2userid> <db2password> <WebSphereAppServerNodeName>**. For example, you would be typing: **SetupUDDI.bat wsdemo wsdemo1 <MYNODE>**, where "MYNODE" is the name of your WebSphere Application Server node name. This batch file will create a JDBC driver and Data source to provide access to the UDDI database, and install the UDDI Registry application.

__31. This batch file copies the UDDI EAR file from the \WebSphere\DeploymentManager\installableApps folder to the \WebSphere\AppServer\installableApps. It also copies the necessary properties files to \WebSphere\AppServer\properties as well as two jar files (uddiresourcebundles.jar and the setupuddimessages.jar) to the \WebSphere\AppServer\lib folder.

__32. The last few messages in your Command Prompt window should look like the following:

```
ADMA5016I: Installation of UDDIRegistry started.
ADMA5005I: Application UDDIRegistry configured in WebSphere
repository
ADMA5001I: Application binaries saved in
C:\WebSphere\AppServer\wstemp\Scriptf1a
2efd3e1\workspace\cells\2KSERVERCC\applications\
UDDIRegistry.ear\UDDIRegistry.ear
ADMA5011I: Cleanup of temp dir for app UDDIRegistry done.
ADMA5013I: Application UDDIRegistry installed successfully.
UDIN2104I: Attempting to save post installation
configuration.
UDIN2105I: Changes saved successfully for UDDI Registry.
UDIN2041I: Starting UDDI application.
UDIN2042I: Application appname started successfully. Value
is: appname=UDDIRegistry
```

NOTE: The UDDI Registry uses a WAS40 Data source so the UDDI Registry can be accessed using an EJB interface. This is for your information only. No specific action is required.

__33. At this point, the UDDI Registry application and database should be successfully installed.

Part Two: Install the MyBank Application into WebSphere Application Server.

__1. Now you are ready to install the MyBank EAR file on the WebSphere V5 Application Server. If the WebSphere Application Server is installed elsewhere, update the value specified in the setupVARs.bat file - see the User Requirement section for specific instructions.

__2. At a Command Prompt, type **CD \LabFiles50\WebServices**.

__3. In the same Command Prompt window, type **InstallMyBankJDBC.bat <db2userid> <db2password> <WebSphereAppServerNodeName>**. For example, you would be typing: **InstallMyBankApp wsdemo wsdemo1 <MYNODE>**, where “MYNODE” is the name of your WebSphere Application Server node name.

__4. The last few lines received from running the JACL script should be similar to the following:

```
Created CMPConnectorFactory for WebSphere Relational Resource
Adapter(cells/BaseApplicationServerCell/nodes/NODENAME:resources.
xml#builtin_rra)
  - Attempting to save new configuration for the Connection
    Factory
  - Changes saved successfully.
```

__5. The setupMyBank.jacl script configures the JDBC Provider, Data source, and Connection Factory for the MyBank Application.

__6. Ensure your path in the Command Prompt still says **\WebSphere\AppServer\bin**.

__7. Type **wsadmin**. When the `wsadmin>` prompt is visible, type one of the following commands:

If you **HAVE** completed Parts 1 and 2 of the Web Services Lab:
$AdminApp install C:/LabFiles50/WebServices/MySolution/MyBank.ear -deployejb

If you **HAVE NOT** completed Parts 1 and 2 of the Web Services Lab:
$AdminApp install C:/LabFiles50/WebServices/Solutions/MyBank_d.ear -deployejb

__8. The above operation takes some time to complete. Be patient. It has to generate the deployed code as part of the application installation.

__9. After the install completes, you should see messages similar to those below.

```
ADMA5005I: Application MyBank configured in WebSphere repository
ADMA5011I: Cleanup of temp dir for app MyBank done.
ADMA5013I: Application MyBank installed successfully.
```

__10. Type **$AdminConfig save**.

__11. Type **exit** in the Command Prompt to exit wsadmin.

__12. Stop the WebSphere Application Server. At a Command Prompt, type **CD \WebSphere\Appserver\bin**. Type **stopserver server1**.

```
C:\WebSphere\AppServer\bin>stopserver server1
ADML0100I: Reading configuration for server: server1
ADML0201I: Server stop request issued. Waiting for stop status.
ADML1000I: Server server1 stop completed.
```

__13. At a Command Prompt, type **CD \WebSphere\AppServer\bin**. Start the WebSphere Application Server by typing **startserver server1**. After successful startup of the application server, you should see resulting console messages like the following:

```
C:\WebSphere\AppServer\bin>startserver server1
ADML0100I: Reading configuration for server: server1
ADML0200I: Server launched. Waiting for initialization status.
ADML0000I: Server initialization completed. Process id is: 252
```

__14. If you **<u>HAVE NOT</u>** completed Parts 1 and 2 of the Web Services lab, complete this step, otherwise proceed to Part Three.

The MyBank application 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 Three: Publish Web Services and Create Sample Application

__1. Populate the UDDI Registry with a Business Entity and Business Service by using a program. This is really a setup step, but since the WebSphere Application Server has to be running with the MyBank application available, this step is placed here.

__ a. At a Command Prompt, **CD \LabFiles50\WebServices\UDDIRegistry\PopulateUDDI**.

__ b. Type **PopulateUDDI.bat wsdemo wsdemo1**. This invokes a Java program that uses the UDDI4J-WSDL APIs to programmatically populate the UDDI Registry with the TransferFunds and CreateAccountWS Web Service descriptions. The source file is located in the same directory as the batch file PopulateUDDI.bat. NOTE: In Part 2 of the Web Services lab, you used Cloudscape as the Unit Test UDDI Registry database in the Application Developer when publishing the Web Service descriptions. Therefore, the Web Service descriptions must be published again. Additionally, there are dependencies on where software has been installed in the PopulateUDDI.bat file. If you have software installed in other locations other than those specified in the User Requirements section of the lab, you may have to update the PopulateUDDI.bat file.

__2. Start WebSphere Studio Application Developer.

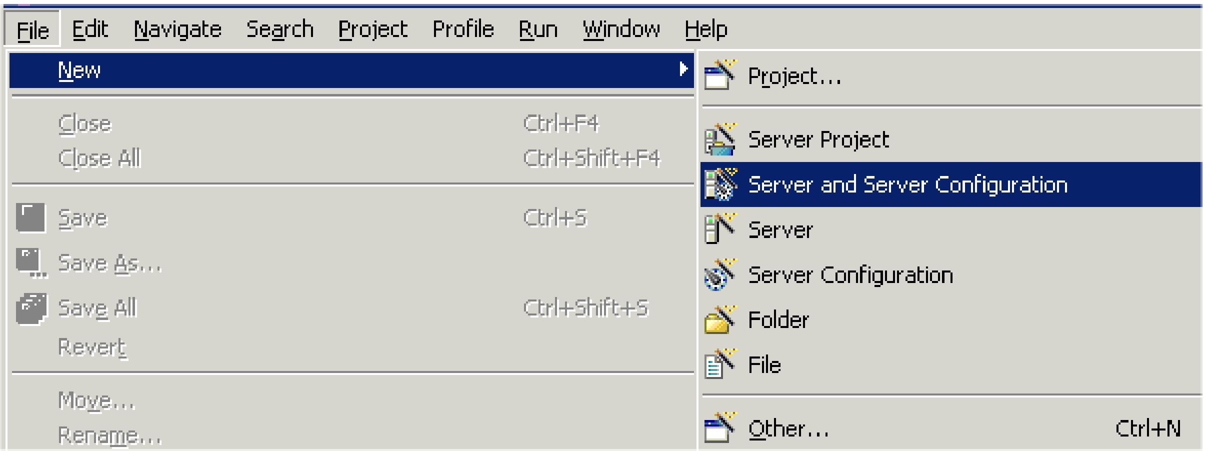
__ a. Select the menu **Start** > **Programs** > **IBM WebSphere Studio** and select **Application Developer 5.0**.

__ 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\testspace** 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.

__ c. Switch to the Server Perspective. Select **Window > Open Perspective > Other....** Select **Server**. Click **OK**.

__3. Create a WebSphere Test Environment Server.

__ a. Select **File > New > Server and Server Configuration**.

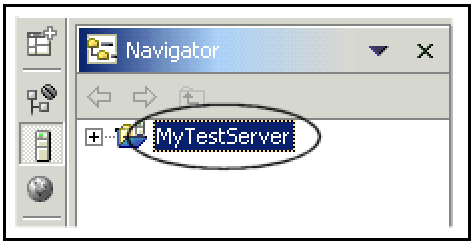


__ b. On the Create a new server and server configuration panel, type **MyTestServer** for the Server name and **MyTestServer** for the folder.

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

__ d. Click **Finish**.

__ e. Select **MyTestServer** in the Navigator View of the Server perspective.

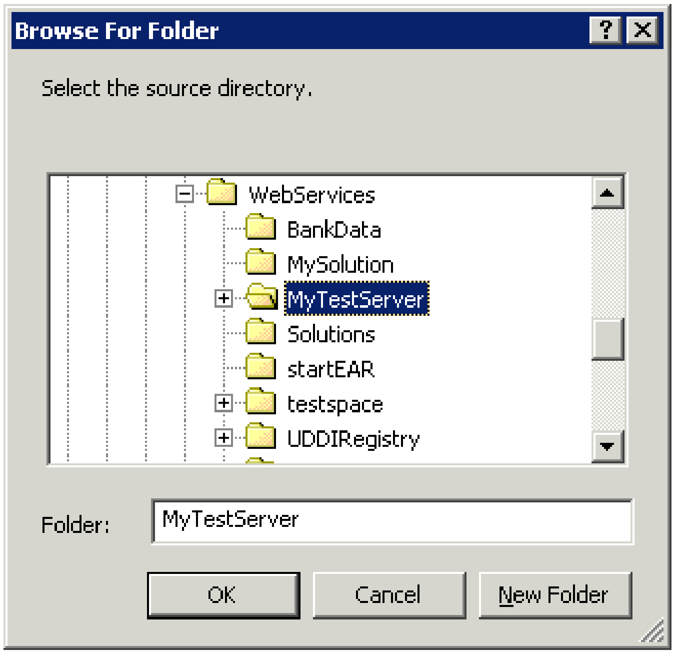


__4. Import a previously defined server configuration into the new MyTestServer server just created. This is done so you avoid modifying all of the port numbers for the server. Recall that the standalone WebSphere Application Server is still running. If the port numbers are not changed the test server within WebSphere Studio Application Developer will not start.

__ a. Select **File > Import... > File System**. Click **Next**.

__ b. Click **Browse...** to the right of Directory on the File system wizard.

__ c. Navigate to **\LabFiles50\WebServices\MyTestServer**. Select the **MyTestServer** folder.

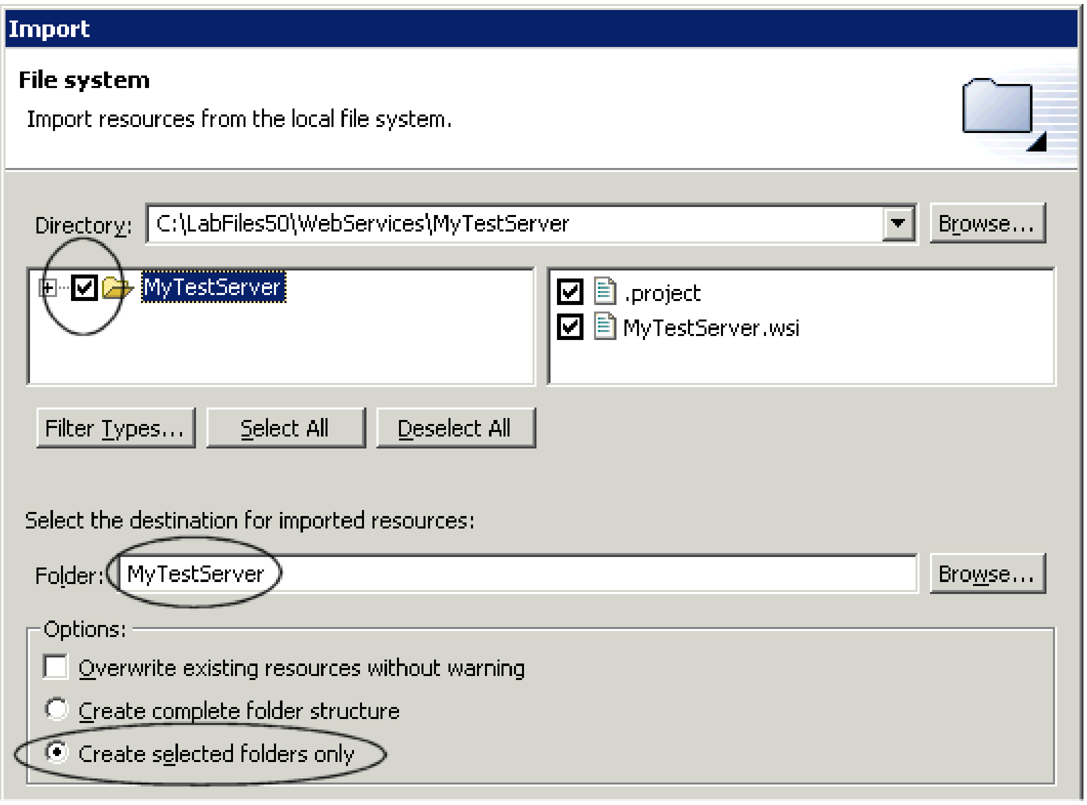


Select the source directory.

Folder: MyTestServer

__ d. Click **OK**.

__ e. Place a **check mark** in the square box next to the MyTestServer folder. Ensure Folder reads **MyTestServer** and select **Create selected folders Only**.



Import

File system

Import resources from the local file system.

Directory: C:\LabFiles50\WebServices\MyTestServer

☑ MyTestServer

☑ .project
☑ MyTestServer.wsi

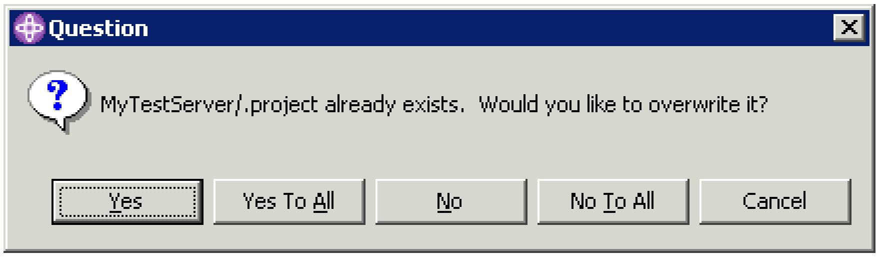
Select the destination for imported resources:

Folder: MyTestServer

Options:
☐ Overwrite existing resources without warning
○ Create complete folder structure
● Create selected folders only

__ f. Click **Finish**.

__ g. Answer **Yes to All** in response to the prompt.



Question

MyTestServer/.project already exists. Would you like to overwrite it?

__5. Create a Web Project to create the Web Service sample application in.

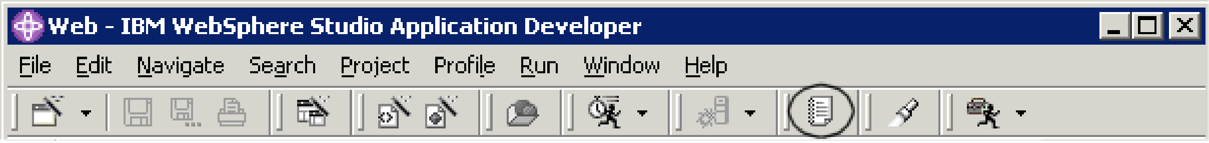
__ a. Switch perspectives to the Web Perspective. Select **Window > Open Perspective > Other....** Select **Web**. Click **OK**.

__ b. Select **File > New > Web Project**.

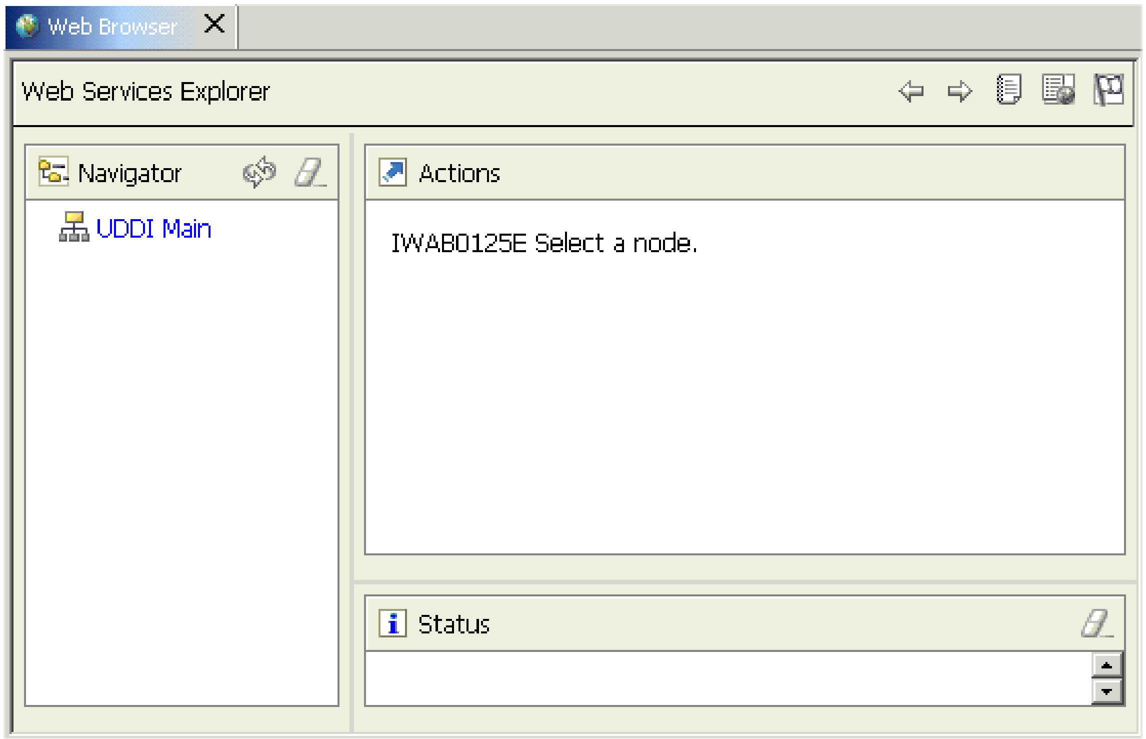
__ c. On the Define the Web Project panel, type **TestWebServicesWeb** for the Project name. Accept the rest of the defaults. Click **Finish**.

__6. Launch the Web Services Explorer to discover the TransferFunds Web Service.

__ a. Click the Toolbar icon that resembles a piece of paper. Place your mouse over the icon and view its popup help to locate the correct icon.



__7. The Web Services Explorer is launched in a browser window and looks like the following graphic. Since the Web Services Explorer was launched from the wizard and there is not a Unit Test UDDI Registry available within our workspace, you must specify the values for the registry name, inquiry and publish URLs.



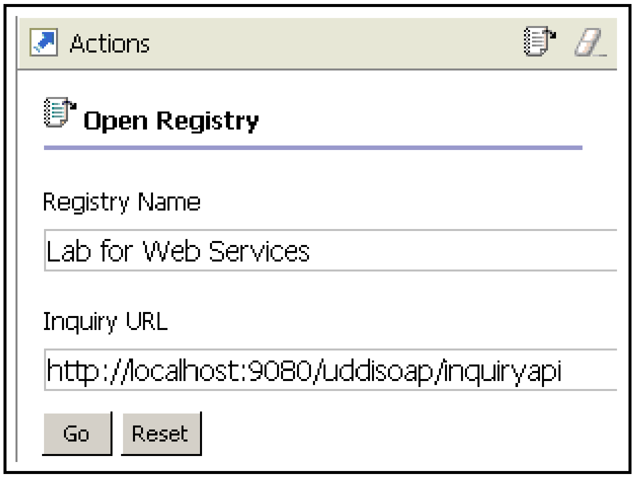
IWAB0125E Select a node.

__8. In the Navigator Frame, select the **UDDI Main** link.

__9. After you click the UDDI Main link, the Actions frame presents to entry boxes for you to enter a Registry Name and an inquiry URL. Enter the following information:

Registry Name: **Lab for Web Services**

Inquiry URL: http://localhost:9080/uddisoap/inquiryapi



Registry Name: Lab for Web Services

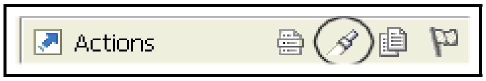
Inquiry URL: http://localhost:9080/uddisoap/inquiryapi

__10. Click **Go**.

__11. In the Status Frame along the bottom of the browser window, you should see a message similar to:

"The registry named Lab for Web Services located at http://localhost:9080/uddisoap/inquiryapi was opened successfully."

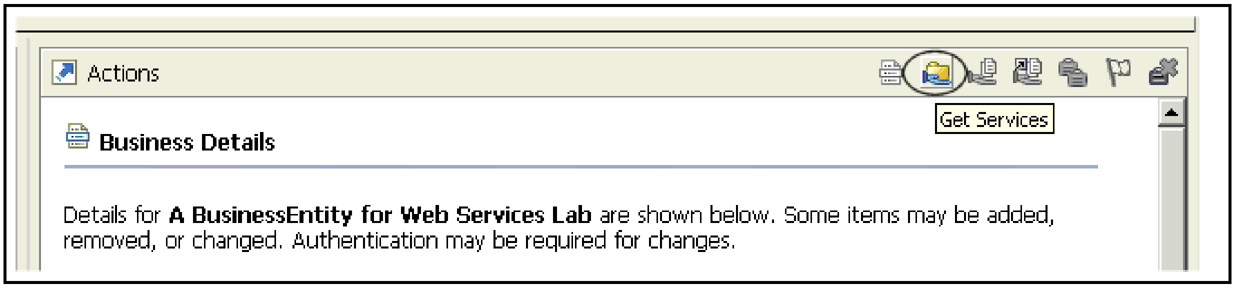
__12. In the **Actions** pane, select the **Find** Action. The icon resembles a flashlight. In the graphic below is it the second icon from the left of the Action Icons.



__13. Select **Businesses** from the Search for list box. Type of search is **Simple**. In the Names field, type **%** (a single percentage character). The '%' character is the UDDI Registry's wild card character.

__14. Click **Go**. This queries the local UDDI Registry (whose database repository is using DB2) and returns all of the Businesses that are registered. If multiple Businesses are registered, multiple rows in a table are presented. In your case, the details for the "**A BusinessEntity for Web Services Lab**" are shown.

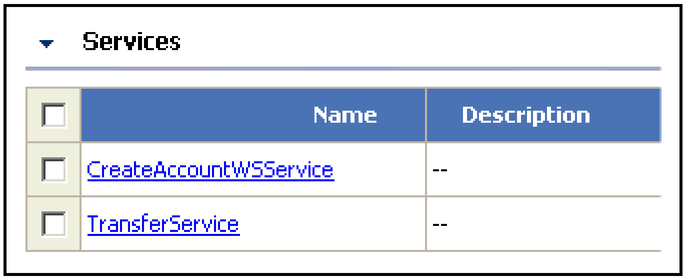
__15. On the Business Details panel, select the **Get Services** icon in the Toolbar.



Business Details

Details for **A BusinessEntity for Web Services Lab** are shown below. Some items may be added, removed, or changed. Authentication may be required for changes.

__16. The Query Results panel shows the services that belong to the Business Entity called "A BusinessEntity for Web Services Lab".

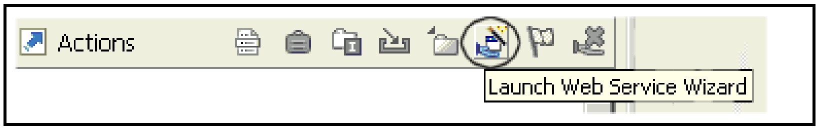


Services

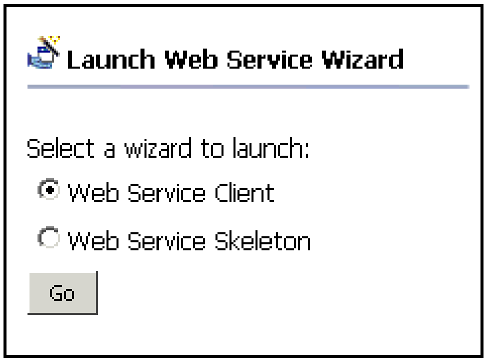
□	Name	Description
□	CreateAccountWSService	--
□	TransferService	--

__17. Select **TransferService** link.

__ a. The Service Details are displayed. Select **Launch Web Services Wizard** icon.



__ b. On the Launch Web Service Wizard, select **Web Service Client**.



Launch Web Service Wizard

Select a wizard to launch:

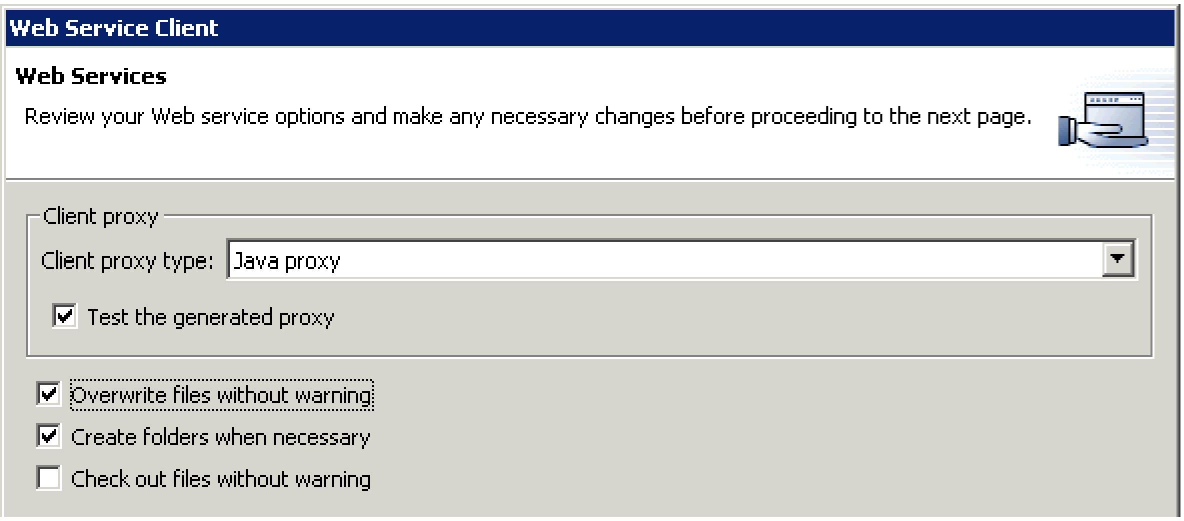
● Web Service Client

○ Web Service Skeleton

Go

__ c. Click **Go**. This launches the Web Service Client wizard within the WebSphere Studio Application Developer.

__ d. On the Web Service Client panel, select **Test the generated proxy**. Ensure **Overwrite files without warning** and **Create folders when necessary** are also selected.



Web Service Client

Web Services

Review your Web service options and make any necessary changes before proceeding to the next page.

Client proxy

Client proxy type: Java proxy

☑ Test the generated proxy

☑ Overwrite files without warning

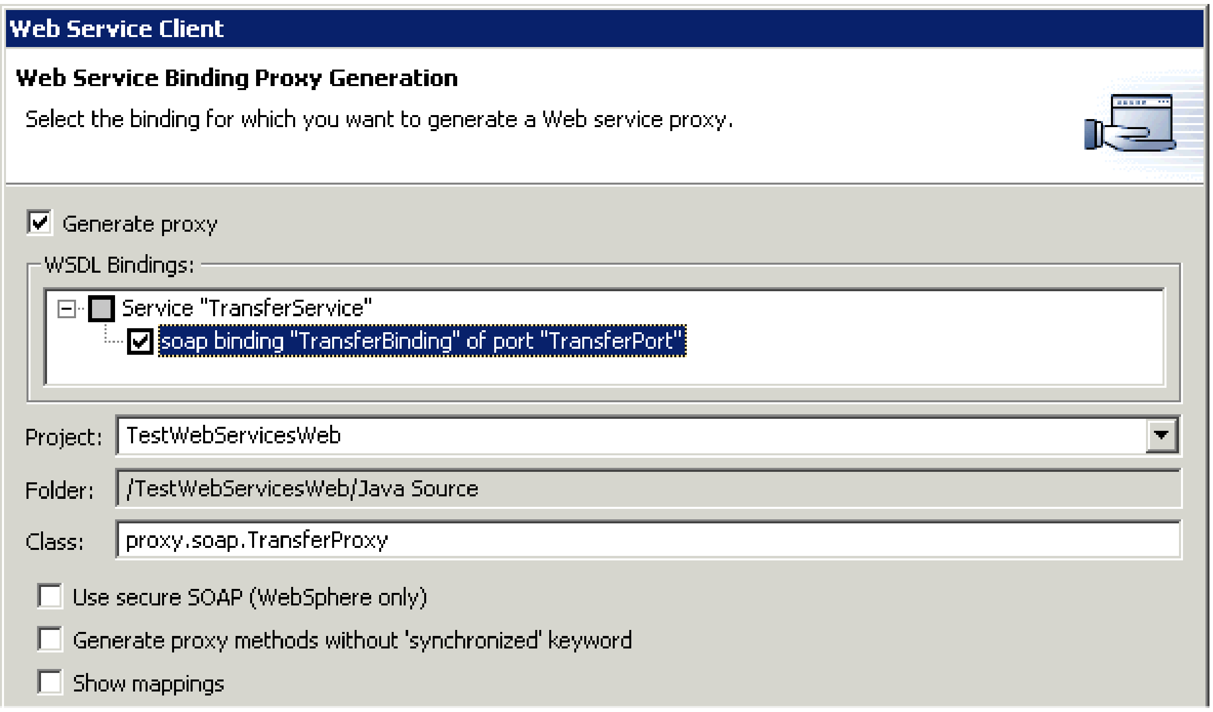
☑ Create folders when necessary

☐ Check out files without warning

__ e. Click **Next**.

__ f. The WSDL file name or URL on the Web Service WSDL File Selection panel is pre-populated with the TransferService WSDL file. Click **Next**.

__ g. On the Web Service Binding Proxy Generation panel, ensure **TestWebServicesWeb** is selected for the Project name. The Folder name will read TestWebServicesWeb/Java Source.



Web Service Client

Web Service Binding Proxy Generation

Select the binding for which you want to generate a Web service proxy.

☑ Generate proxy

WSDL Bindings:

- □ Service "TransferService"
  - ☑ soap binding "TransferBinding" of port "TransferPort"

Project: TestWebServicesWeb

Folder: /TestWebServicesWeb/Java Source

Class: proxy.soap.TransferProxy

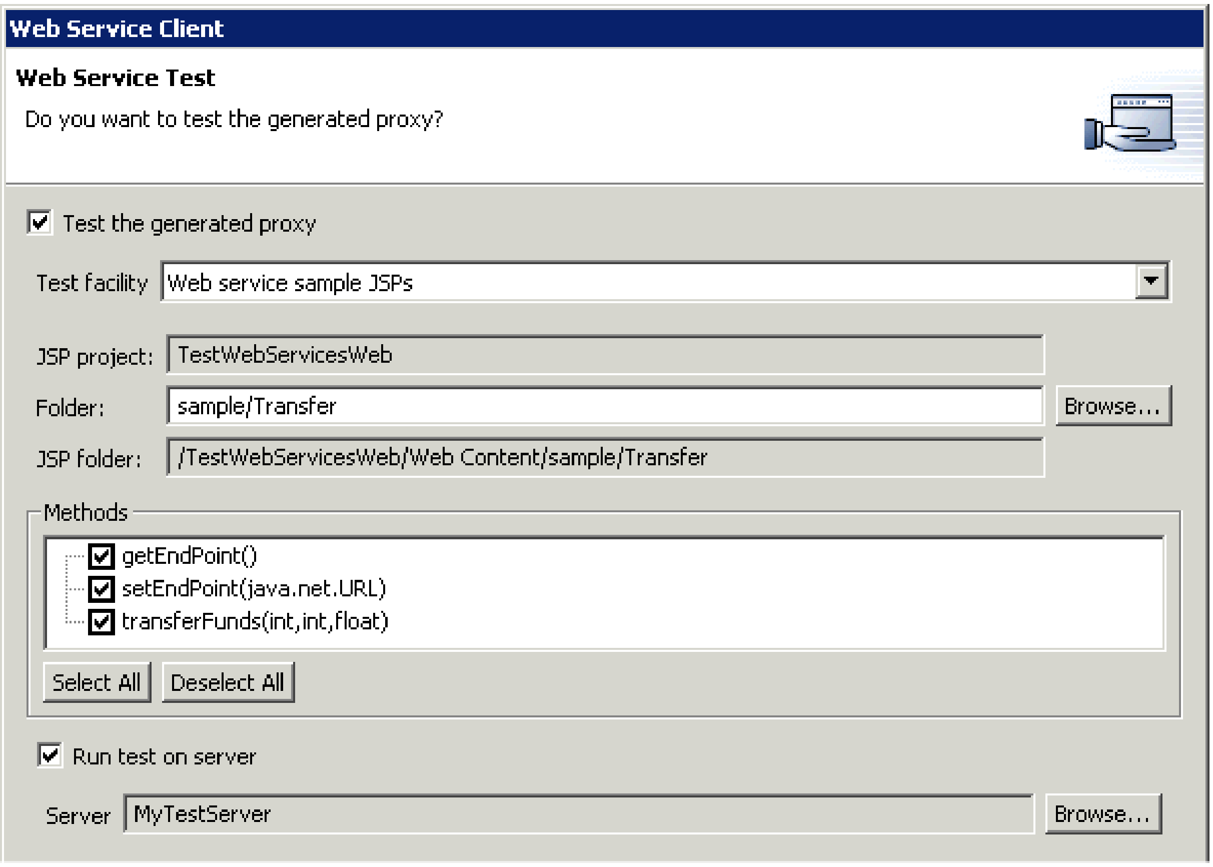
□ Use secure SOAP (WebSphere only)

□ Generate proxy methods without 'synchronized' keyword

□ Show mappings

__ h. Click **Next**.

__ i. On the Web Service Test panel, ensure the **Test the generated proxy** option is selected. Accept the remaining defaults.



Web Service Client

Web Service Test

Do you want to test the generated proxy?

☑ Test the generated proxy

Test facility: Web service sample JSPs

JSP project: TestWebServicesWeb

Folder: sample/Transfer

JSP folder: /TestWebServicesWeb/Web Content/sample/Transfer

Methods

☑ getEndPoint()

☑ setEndPoint(java.net.URL)

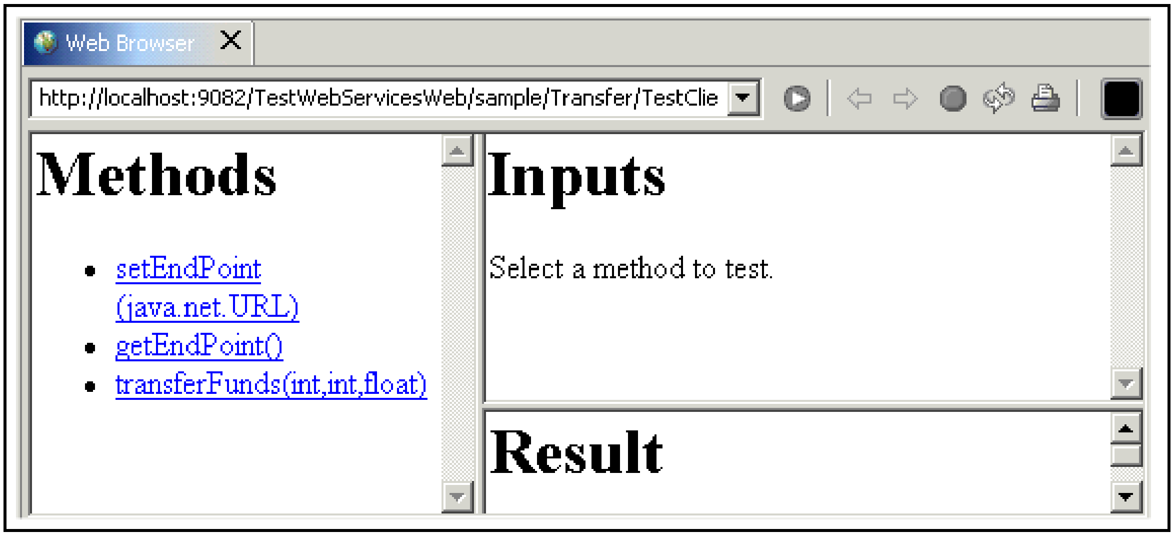
☑ transferFunds(int,int,float)

☑ Run test on server

Server: MyTestServer

__ j. Click **Finish**.

__18. The Web Services wizard creates the sample application and starts the “MyTestServer” server as well. The following sample application (TestClient.jsp) is generated. By selecting the transferFunds method, you can test the execution of this Web Service. The desired outcome is to transfer funds between two accounts in the DB2 Database.



http://localhost:9082/TestWebServicesWeb/sample/Transfer/TestClie

Methods

- setEndPoint (java.net.URL)
- getEndPoint()
- transferFunds(int,int,float)

Inputs

Select a method to test.

Result

__19. Verify the balances of the accounts in the database before we transfer funds between two accounts.

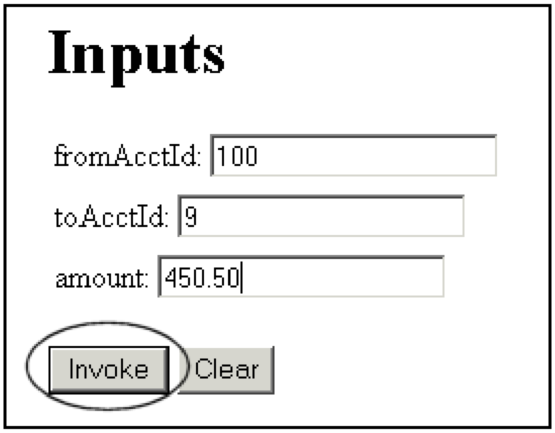
__20. At a Command Prompt, **CD \LabFiles50\WebServices\Solutions**. Type **CheckAccountBalance.bat wsdemo wsdemo1**. If you did not complete Parts 1 and 2 of the Web Services lab, use an account number other than 100 in the following steps. The result of this batch file should be as follows:

```
ACCOUNTNUMBER BALANCE
------------- -----------

            1     1000.00
            2     2000.00
            3     3000.00
            4     4000.00
            5     5000.00
            6     6000.00
            7     7000.00
            8     8000.00
            9     9000.00
           10    10000.00
          100     5000.50

  11 record(s) selected.
```

__21. Transfer 450.50 from Account #100 to Account #9. Note the beginning Account balances are 5000.50 and 9000.00 respectively.

__ a. In the Method frame, select **transferFunds**.

__ b. In the Inputs frame, type **100** for the fromAcctId, **9** for the toAcctId, and **450.50** for the amount. Click **Invoke**.



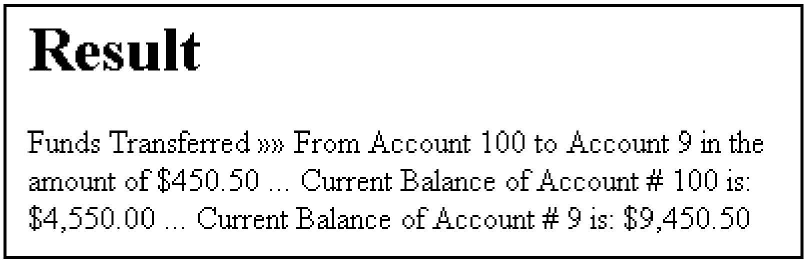
Inputs

fromAcctId: 100

toAcctId: 9

amount: 450.50

__ c. The Result frame should read as follows.



Result

Funds Transferred »» From Account 100 to Account 9 in the amount of $450.50 ... Current Balance of Account # 100 is: $4,550.00 ... Current Balance of Account # 9 is: $9,450.50

__ d. To verify the correct result, you will run the database query again. At a Command Prompt, **CD \LabFiles50\WebServices\Solutions**. Type **CheckAccountBalance.bat wsdemo wsdemo1**. The result of this batch file should be as follows:

```
    ACCOUNTNUMBER BALANCE
    ------------- -----------
    1        1000.00
    2        2000.00
    3        3000.00
    4        4000.00
    5        5000.00
    6        6000.00
    7        7000.00
    8        8000.00
    9        9450.50 <<--- updated
   10       10000.00
  100        4550.00 <<--- updated
11 record(s) selected.
```

__22. Stop the Test Environment server.

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

__23. Close the WebSphere Studio Application Developer application.

Part Four: Cleaning up the lab machine

__1. At a Command Prompt, **CD \WebSphere\AppServer\bin**. Type **wsadmin**. At the prompt, type **$AdminApp uninstall MyBank**. This will uninstall the MyBank Application. After the uninstall completes, type **$AdminConfig save**. Type **exit** in the command prompt after the uninstall completes.

__2. Stop the WebSphere Application Server. At a Command Prompt, **CD \WebSphere\AppServer\bin**, then type **stopserver server1**.

Installing the Solution(s)

__1. In the \LabFiles50\WebServices\Solutions sub-directory there is an EAR file called MyBank_d.ear that can be used to install directly in the WebSphere Application Server V5. After installation of the application, you can type the following into a web browser: http://localhost:9080/MyBankWeb/sample/CreateAccountWS/TestClient.jsp.

__2. For discovering a Web Service and creating a sample application, follow the steps outlined in Part Three of this lab.

What you did in this exercise

In this lab exercise, you installed the MyBank J2EE application into the WebSphere Application Server runtime environment. You also installed the UDDI Registry that is packaged with WebSphere Network Deployment. After programmatically publishing the Web Services to the UDDI Registry using the UDDI4J-WSDL APIs, you created a sample application to invoke the Web Service.

As a Service Requester, you used the Web Services explorer (from WebSphere Studio Application Developer) to discover the TransferService and then used the Web Service client wizard to generate a sample application. Running the sample application showed the Web Service working successfully.