IBM WEBSPHERE 6.0 – LAB EXERCISE

WebSphere Rapid Deployment – Deployment Automation

Note: Education materials and other documentation as applicable including programming manuals, operating guides, physical planning manuals and installation manuals related to the IBM Products may be early versions subject to change. Documents will be furnished solely for the purpose of and for the duration of the Beta Test.

What this exercise is about

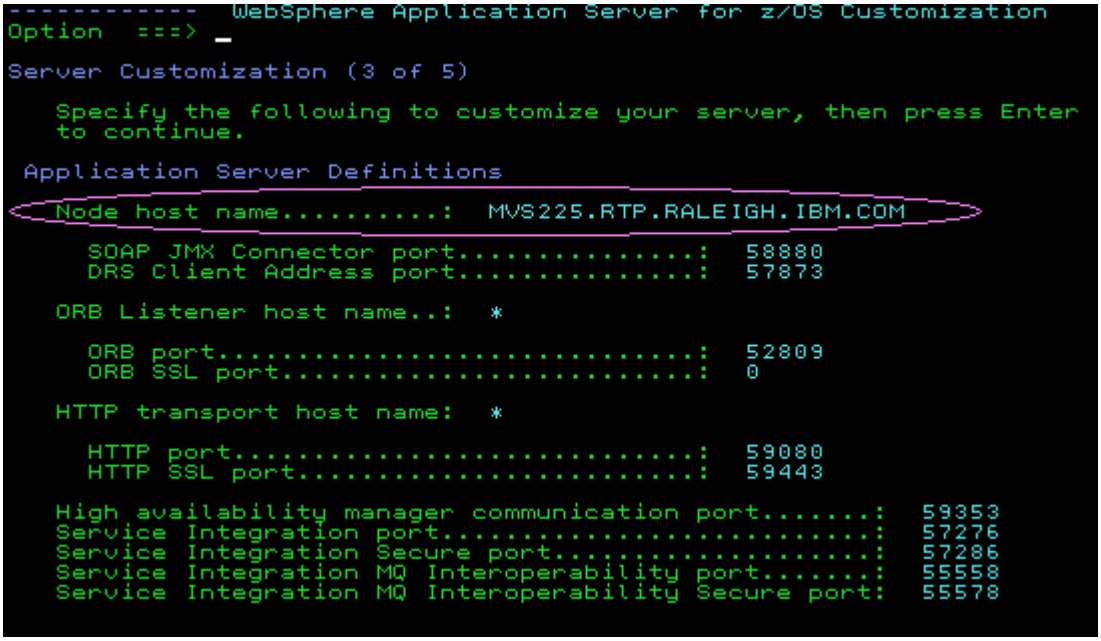
The objective of this lab is to provide an understanding of WebSphere Rapid Deployment’s (WRD) deployment automation features. WRD gives administrators and developers a simpler and faster way to develop, deploy, construct, install, update, and uninstall a J2EE application. By configuring a “watched” directory within the file system, installing an enterprise application is as easy as dropping the EAR file into a folder.

This lab will entail configuring two styles of deployment automation. The first, called autoappinstall, will automatically install, update, and uninstall an enterprise application on WebSphere. You will add the file to the monitored directory, which will cause an install, then overwrite the file, which will cause an update, and finally remove the file which will cause an uninstall. In the final part of this lab you will construct another WRD deployment automation project. However, this new project will use the freeform style. Using this style you will place simple J2EE artifacts, like servlets, jsps, etc, inside the directory and WRD will automatically place those file in the correct J2EE project structure and install the application for you. You will then make updates by overwriting the individual J2EE artifacts instead of the having to overwrite the entire web module or enterprise application.

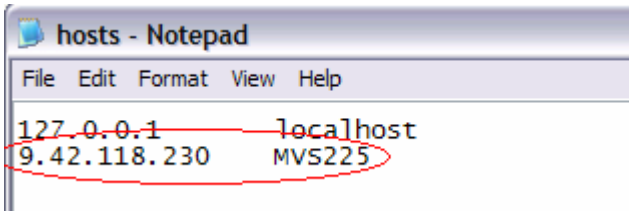
Lab Requirements

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

- WebSphere Application Server v6.0.1 should be installed.
- Installation of lab sample code into the following directories:
  - Windows workstation: **C:\Labfiles60**
  - z/OS host system: **/etc/Labfiles60**
- Experience with previous versions of WebSphere Application Server and the J2EE programming model are also required.
- Completion of lab: **Installing WebSphereBank Application using Cloudscape** or **Application Update.**
- Also, you should verify that the Node host name is set to a value that is resolvable by the machine on which you are running WRD.



If a short host name is used, such as MVS225, you will need to update you hosts file on your workstation (C:\WINDOWS\system32\drivers\etc\hosts) with the ip address of the remote system:



What you should be able to do

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

- Configure WebSphere Rapid Deployment for the autoappinstall deployment automation style.
- Configure WebSphere Rapid Deployment for the freeform deployment automation style.

Introduction

The J2EE 1.4 application you will be using is a simple banking application named WebSphereBank. It is composed of an EJB, Web, and Application Client module. In the EJB module there is a session bean and an two entity beans. Before being able to fully use the application, you will need to create the database and tables required to run the application and create the datasource resource in WebSphere Application Server so your application can persist data to the database. You will be using a Cloudscape database which is a small footprint, object-relational database based purely on Java. It can be set up very easily and you can quickly test your application to ensure it is working properly. The EJB to RDB mappings and the deploy code have already been generated for you.

In the first part of the lab you will create a WRD deployment automation project using the autoappinstall style. This will create a monitored directory on your file system. Once it's configured you will place the WebSphereBank EAR file in the directory and verify that the application was installed by using the administrative console and accessing the application itself. However, this version of WebSphereBank will not have any functionality to perform a transfer of funds or a retrieve balances. The next step will be to take a new WebSphereBank EAR file with transfer and retrieve funds function and overwrite the old WebSphereBank EAR file. Again, you will verify that the application did get updated. The final step of the autoappinstall style will be to remove the file from the directory and check through the administrative console that the application did indeed, get uninstalled. In the final section of this lab you will create another WRD deployment automation project using the freeform style. You will be using a simple web application that contains one servlet. This level of simplicity will show you the basics of how the freeform style works.

Exercise Instructions

Some instructions in this lab may be Windows operating-system specific. If you plan on running the lab on an operating-system other than Windows, you will need to execute the appropriate commands, and use appropriate files(.sh vs. .bat) for your operating system. The directory locations are specified in the lab instructions using symbolic references, as follows:

Location Reference	Windows example	z/OS example
<WAS_HOME>	C:\WebSphere\AppServer\	/etc/c6cellB/AppServer
<HOST_NAME>	localhost	mvs22x.rtp.raleigh.ibm.com
<SERVER_NAME>	Server1	C6Server01

<HOST_LAB_FILES>	/etc/LabFiles60
<LOCAL_LAB_FILES>	C:\LabFiles60

Part 1: Preparing the Environment on Host System

____ 1. You will prepare the lab setup script.

- Telnet to your host system:

  telnet MVS22x.RTP.RALEIGH.IBM.COM 1023

  cd /etc/LabFiles60/common

- Within **<HOST_LAB_FILES>/common** directory, edit **labSetup.sh** file to ensure that the variables are correct for WAS_HOME, WAS_USER_SCRIPT, and LAB_FILES. Save the above changes and close the file. If you are running this lab on the class systems, all the information should be correct. So if you don't feel comfortable using vi editor, you can skip this step.

  vi labSetup.sh

  "j" scrolls down.

  ":q" quits and ":wq" writes and quits

____ 2. Issue the following commands to ensure the scripts have "execute" rights:

  chmod +x labSetup.sh

____ 3. Ensure the server is started

- At the command prompt, navigate to **<WAS_HOME>/profiles/default/bin**

  cd /etc/c6cellB/AppServer/profiles/default/bin

- Check to see if the server is running

  ./serverStatus.sh <SERVER_NAME>

- If the server status does not indicate STARTED, then start the server:

  ./startServer.sh <SERVER_NAME>

Part 2: Creating the BANKDB Cloudscape database

This step demonstrates the creation of the Cloudscape BANKDB database, which you will need for the WebSphereBank application. You will use commands to create the database and tables. The database will be created in the <LAB_FILES>/CloudscapeDB directory.

____ 4. Generate the Cloudscape BANKDB database and tables.

__ a. In a command window, navigate to

/etc/c6cellB/AppServer/cloudscape/bin/embedded

__ b. Issue the Cloudscape **ij** command to start the Cloudscape utility

./ij.sh

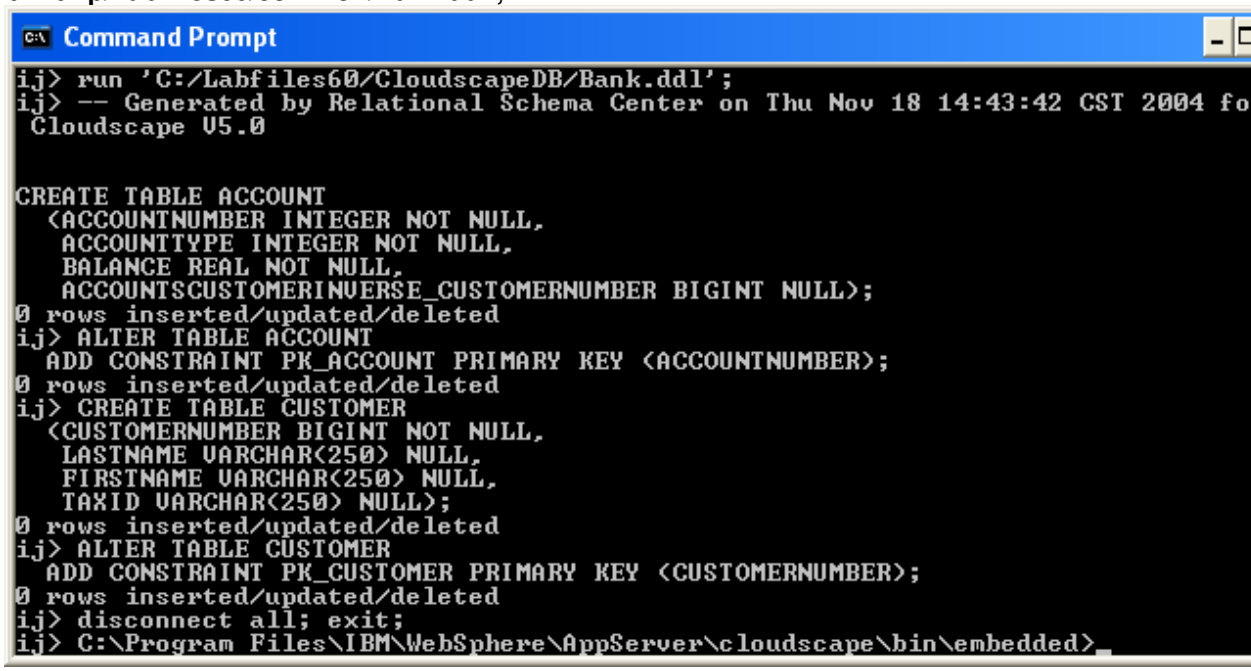
__ c. Issue the following commands, replacing <LAB_FILES> with your lab file path:

mkdir /tmp/LabFiles60/CloudscapeDB

connect 'jdbc:db2j:/tmp/LabFiles60/CloudscapeDB/BANKDB;create=true';

Note: this command may run for 10 to 30 seconds; when the command completes, you will see the "ij>" prompt with no messages

1) **run 'tmp/LabFiles60/common/Bank.ddl';**



2) disconnect all; exit;

3) Go to the <LAB_FILES> directory and issue 'chmod -R 777 CloudscapeDB'

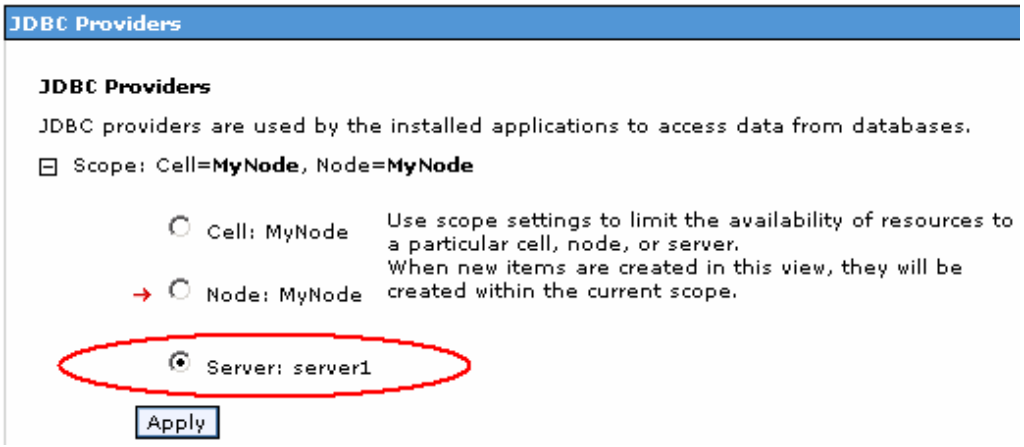
The BANKDB database and tables have been created.

Part 3: Configure a JDBC Driver and a Data Source

You will define and configure the JDBC Driver and Data Source so the WebSphereBank application can access the Cloudscape BANKDB database.

____ 1. Data sources and other resources are organized under the Resources heading. Expand **Resources** and click **JDBC Providers** to configure the data source which the WebSphereBank application will use to store data.

____ 2. The existing JDBC Providers that are configured at the Node level will be displayed. In order to configure a new provider scoped to the Application Server, select **Server: <SERVER_NAME>**.



____ 3. Click **Apply**

Note: the JDBC Providers that are configured at the Server level were not displayed immediately when you clicked <SERVER_NAME>; they were only displayed when you clicked Apply.

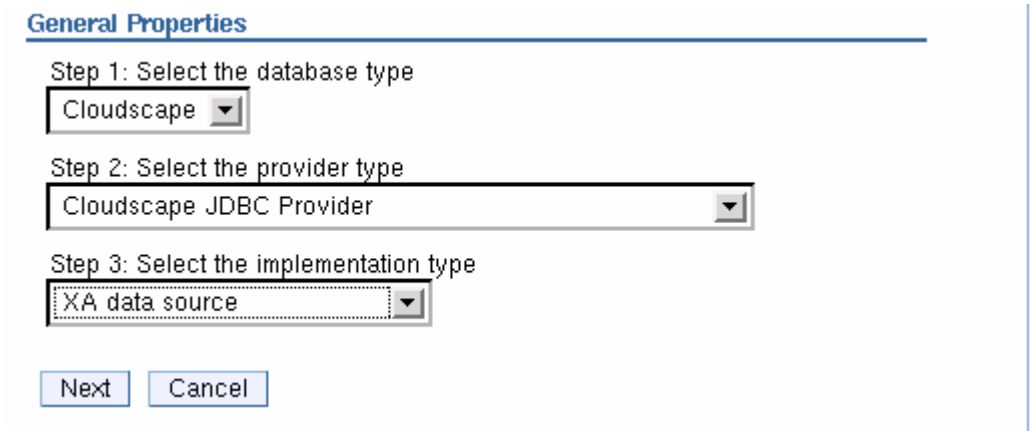
Note: if the node is federated with a Deployment Manager, you will see a browse button and will need to browse to the node before selecting a server.

____ 4. Click **New**. In the **General properties** screen, select:

__ a. Step1: Select the database type as **Cloudscape.**

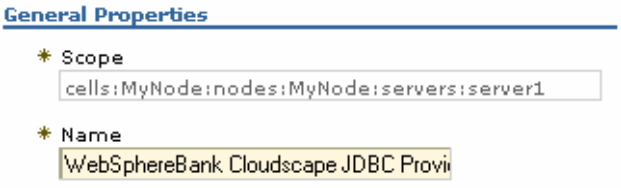
__ b. Step 2: Select the provider type as **Cloudscape JDBC Provider.**

__ c. Step 3: Select the implementation type as **XA data source**.

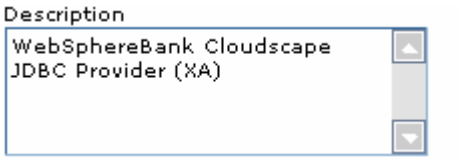


____ 5. Click **Next**.

____ 6. In this screen change the provider name to **WebSphereBank Cloudscape JDBC Provider (XA)**



____ 7. Change the description to WebSphereBank Cloudscape JDBC Provider (XA)



____ 8. Click **OK**. The list of JDBC Providers configured at the server scope will be displayed.

____ 9. Click **Save** in the banner at the top of the page.

____ 10. Click the **Save** button.

____ 11. Click **WebSphereBank Cloudscape JDBC Provider (XA)**

____ 12. Click **Data Sources** under Additional Properties on the right side of the window.

____ 13. Click **New** to create a Data Source.

____ **14.** The empty configuration properties for the data source will be displayed. Enter the following:

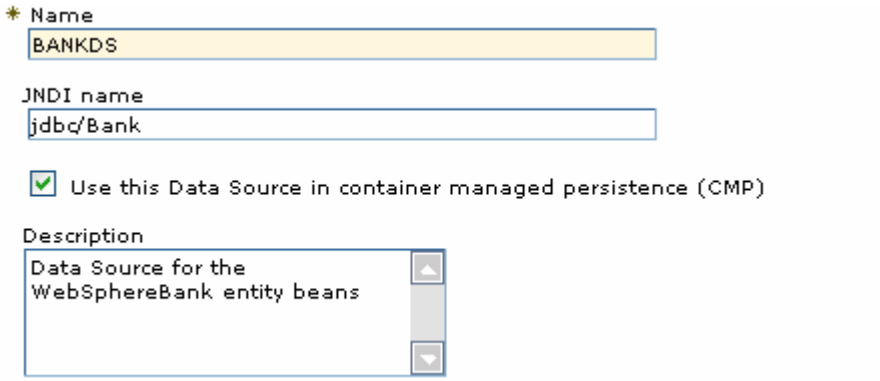
__ a. Name: **BANKDS**

__ b. JNDI name**:** **jdbc/Bank**

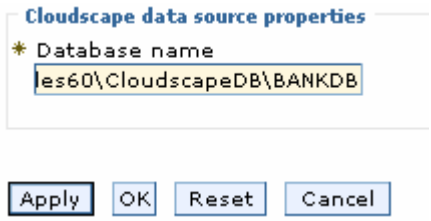
__ c. Make sure there is check next **to "Use this Data Source in container managed persistence (CMP)"**. .

Note: If this box is not checked, you will have problems at runtime because the corresponding connection factory will not be created.

__ d. Description: Data Source for the WebSphereBank entity beans



__ e. Near the bottom of the same screen under Cloudscape data source properties type the **Database name** as **<HOST_LAB_FILES>/CloudscapeDB/BANKDB** (Ex. /etc/LabFiles60/CloudscapeDB/BANKDB):



____ 15. Click **OK**; the data source BANKDS will appear within the list of data sources.

____ 16. Click **Save** in the banner at the top of the page.

____ 17. Click the **Save** button.

____ 18. Click **BANKDS.**

____ 19. On the right hand side click on **Custom properties.**

__ a. Click **connectionAttributes.**

__ b. Enter the value **upgrade=true.**

Note: The value “upgrade=true” will cause Cloudscape database version to be checked. If it is at an earlier version level, the database will be converted to the current level.

__ c. Click **OK.**

____ 20. Your Data Source is completely defined. With WebSphere Application Server v6.0, configuration changes to your server, such as defining a JDBC Provider and Data Source, must be saved with a explicit call to the Save operation in the Administrative console.

____ 21. Click **Save** in the banner at the top of the page.

____ 22. Click the **Save** button.

____ 23. Log out of the Admin Console.

__ a. Click the **Logout** link in the header.

____ 24. Stop and restart the server from you telnet session.

__ a. Stop the server

```
./stopServer.sh <SERVER_NAME>
```

__ b. Start the server

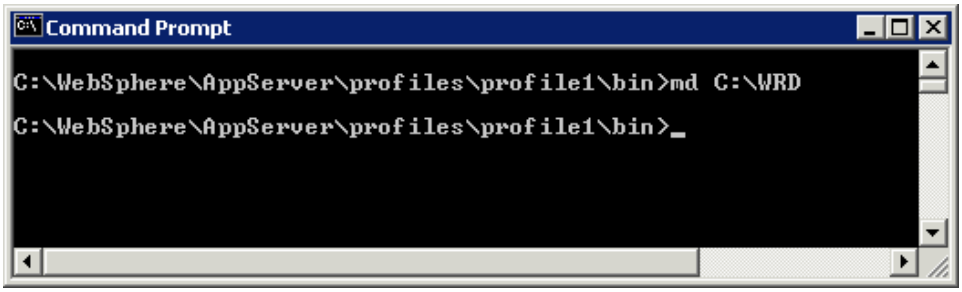
```
./startServer.sh <SERVER_NAME>
```

Note: There are two reasons for stopping and restarting the application server at this point. The first is that changes to Data Sources are picked up at startup. The second is that there is a naming issue in the Beta with EJBs not being removed from the namespace when the application stops. If the Samples had been installed, installing and starting WebSphereBank at this point would result

Part 4: WRD Deployment Automation "autoappinstall" style: WebSphereBank

In this part you will set up a WRD workspace that supports the Auto Application Install style. You will add an EAR file to the project, update it, and then remove it to see how the changes are reflected on the application server. The *automatic application installation* configuration creates a single monitored directory that listens for fully composed EAR files or application module such as WAR, EJB jars, or stand-alone RAR files. If you place EAR files inside this monitored directory, the EAR is automatically deployed to the server. If you delete the EAR file from the monitored directory, then that application is uninstalled from the server. If you place WAR or EJB jar files in the monitored directory, the rapid deployment tool generates the necessary EAR wrapper and then publishes that EAR file on the server. For RAR files, a wrapper is not created. The stand-alone RAR files are published to the server.

____ 1. In the command prompt on your work station, type **md C:\WRD** to create a folder called WRD which will hold the WRD projects that will be defined.

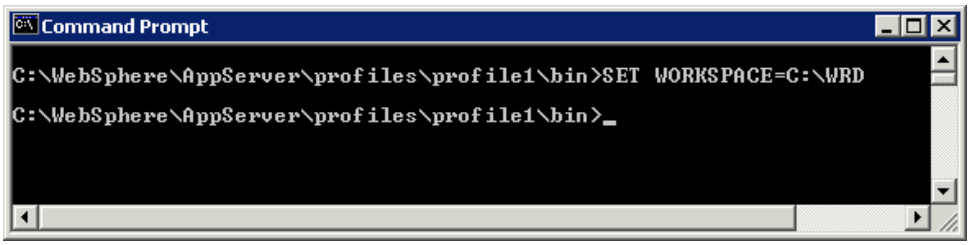


____ 2. Navigate to **C:\IRAD60\runtimes\base_v6\bin** if not already there.

```
cd \IRAD60\runtimes\base_v6\bin
```

____ 3. Configure the WRD Deployment Automation project using the "autoappinstall" style.

__ a. At the command prompt type **SET WORKSPACE=C:\WRD** and press enter. This will set the WRD workspace directory to the folder created in step 1.



__ b. At the command prompt type **wrd-config.bat -project "AutoInstall" -style "autoappinstall"** and press **Enter**:

Note: -project is the name of your WRD deployment automation project and –style is the style that applies to the project

---

__ c. This command will execute for a minute as it starts to build the project. It will then ask for configuration parameters.

1) Fill in the server name: ex. G6Server1

2) Fill in JMX host name: ex. mvs22x.rtp.raleigh.ibm.com

3) Fill in JMX port: ex. 8880

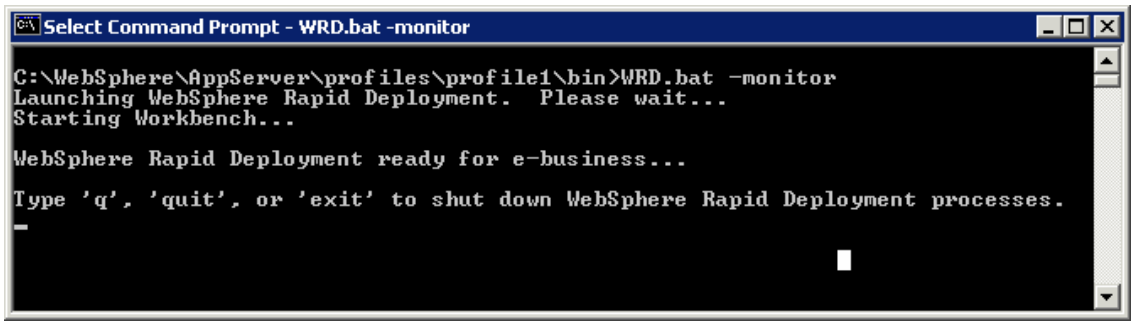
4) Accept the defaults for the rest of the parameters by pressing enter after each one.

```
Initializing the workspace...

------------------------------------------------------------
Parameter Configuration Settings
------------------------------------------------------------
Press ENTER to accept defaults
The * symbol denotes required input

 Enter the server name* ( server1 ) : C6Server01
 Enter the server JMX host name* ( localhost ) : mvs225.rtp.raleigh.ibm.com
 Enter the server JMX port number* ( 8880 ) : 8880
 Enter a path for containing the exported EAR ( -- ) :
 Enter your server username ( -- ) :
 Enter your server password ( -- ) :

Configuring the workspace...
Building the workspace...
WebSphere Rapid Deployment configuration completed.

C:\IRAD60\runtimes\base_v6\bin>_
```

__ d. Once the configuration of the project is complete a confirmation message “WebSphere Rapid Deployment configuration completed” will appear.

____ 4. Start the WRD project.

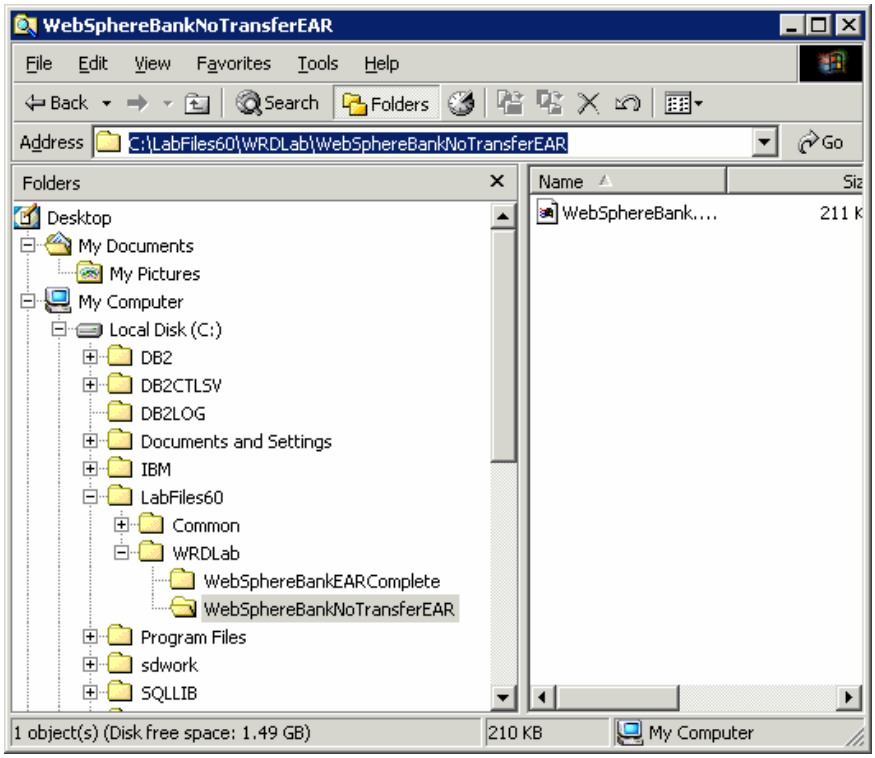
__ a. At the command prompt type: **WRD.bat –monitor** and press **Enter.** The command may take a minute to process. After it’s complete the following output will appear:



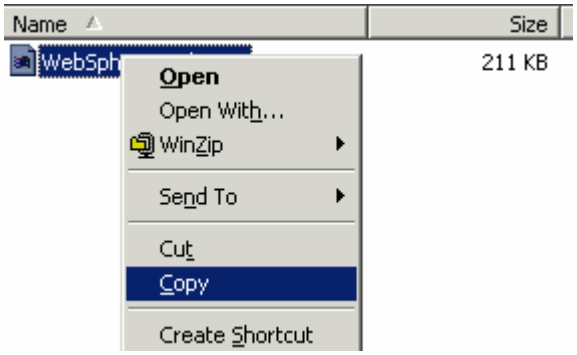
Note: Specifying the –monitor parameter will allow the console output to be directed at the command prompt window to detect error, success, and overall status messaging. A log file can also be used.

____ 5. Install an application using the WRD Auto App Install project just created.

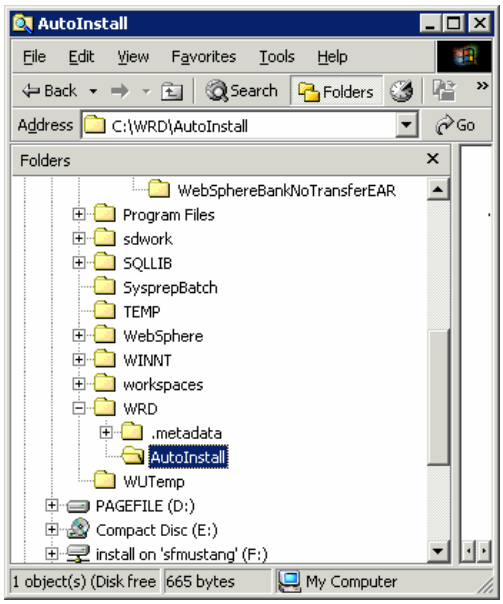
__ a. Open Windows Explorer and navigate to **C:\LabFiles60\WRDLab\WebSphereBankNoTransferEAR**



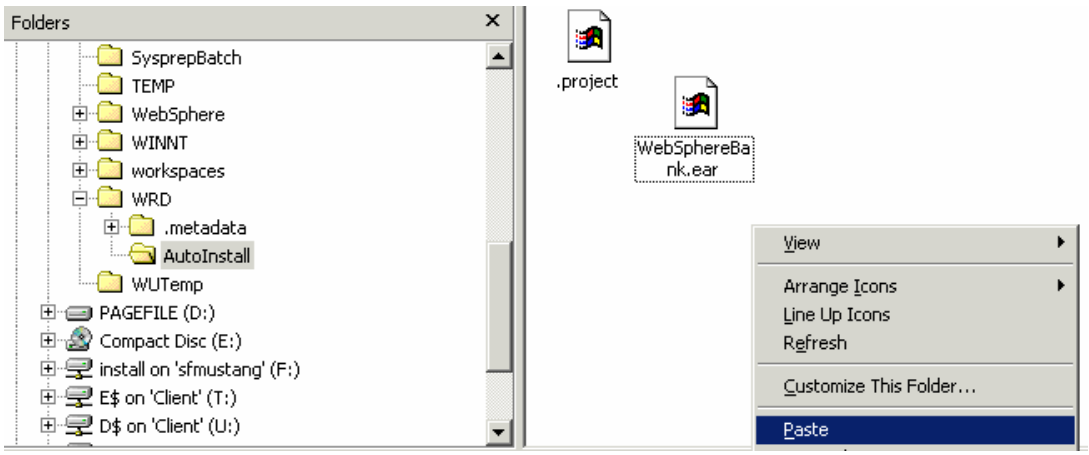
__ b. **Right-click** on the file WebSphereBank.ear and choose **Copy.**



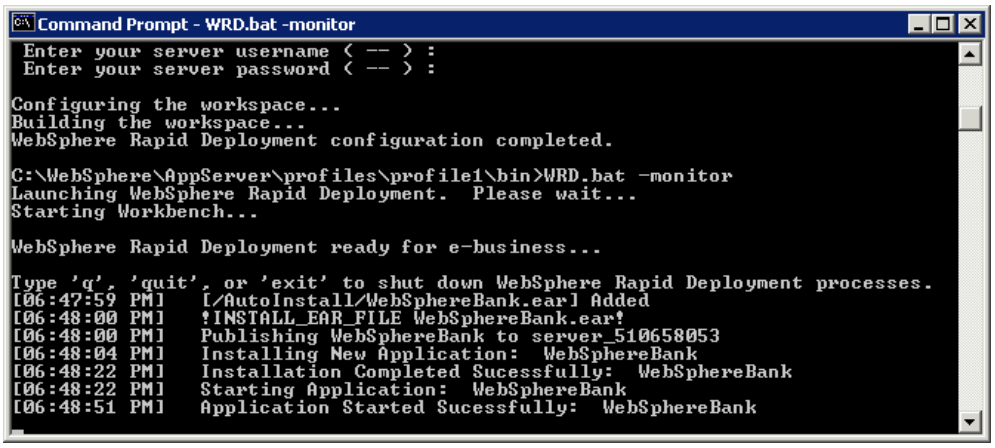
__ c. Staying in Windows Explorer, navigate to **C:\WRD\AutoInstall** directory.



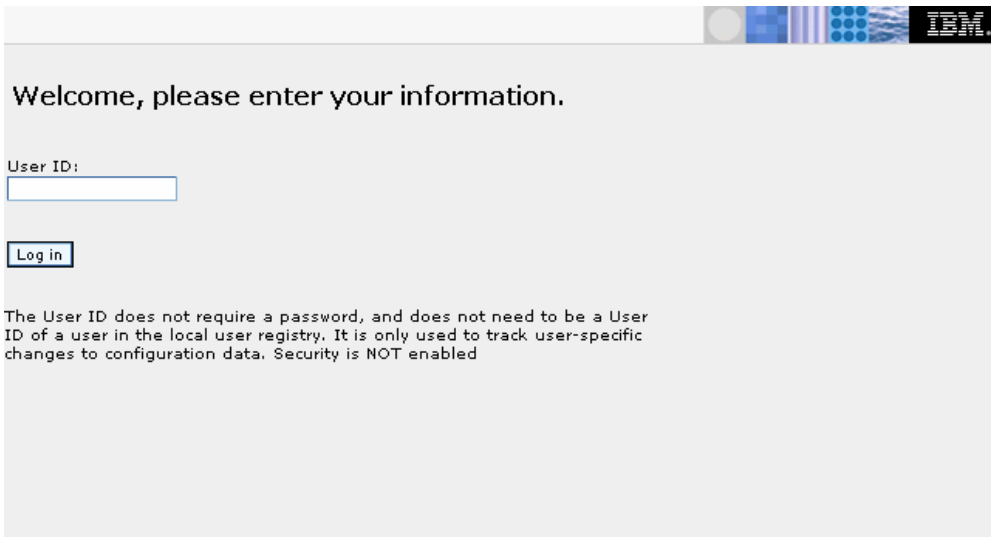
__ d. **Paste** the **WebSphereBank.ear** file copied in step b above to the **C:\WRD\AutoInstall** directory.



__ e. Check the command prompt where WRD was started, it should resemble the screen capture below.



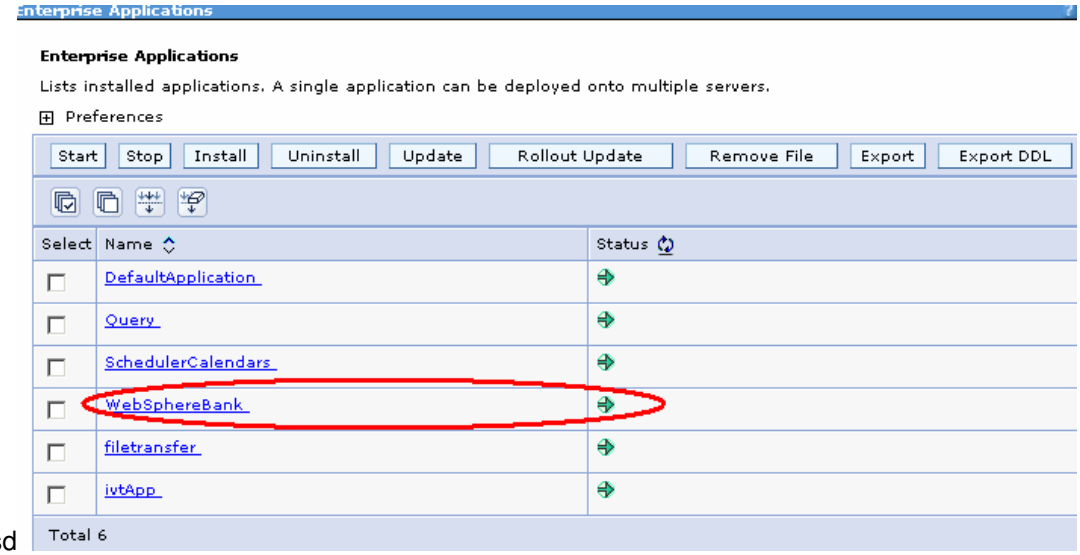
__ f. Open the web browser and navigate to the administrative console using the URL: http://<HOST_NAME>:9080/ibm/console



__ g. Type in **wsdemo** for the User ID and click on **Log in.**

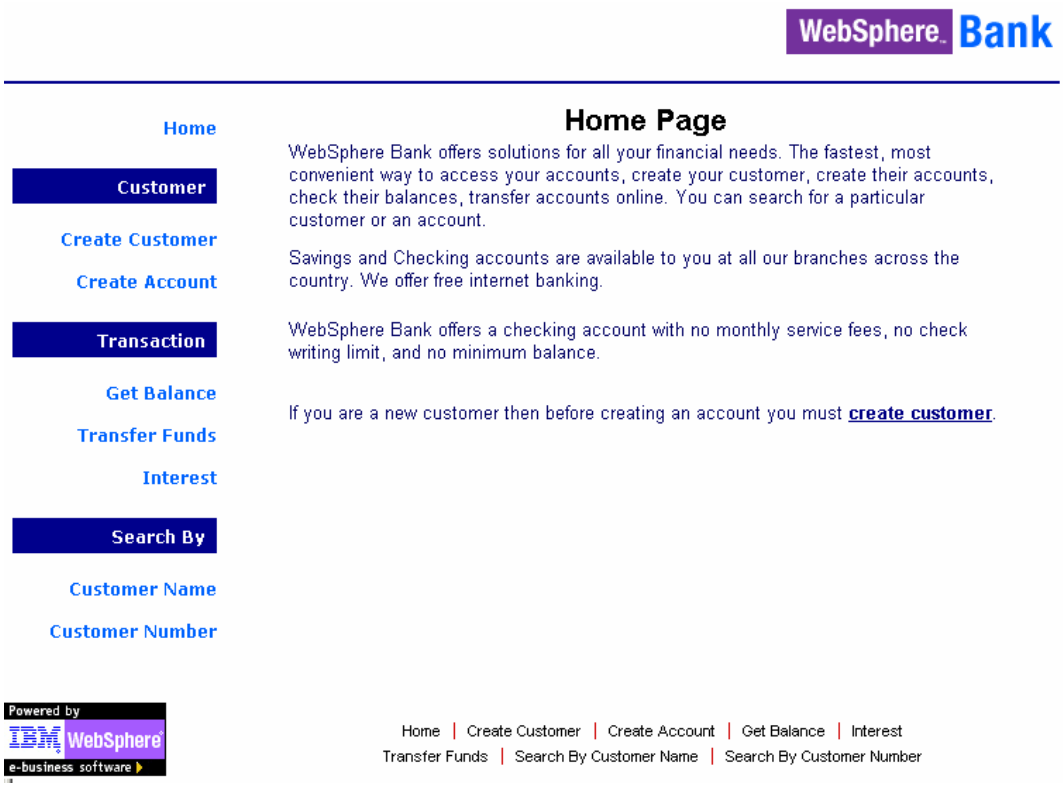
__ h. Expand **Applications** and click on the **Enterprise Applications** link.

__ i. Notice the **WebSphereBank** application is installed and running.

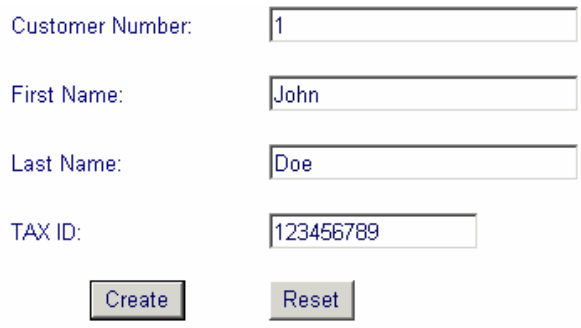


wsd

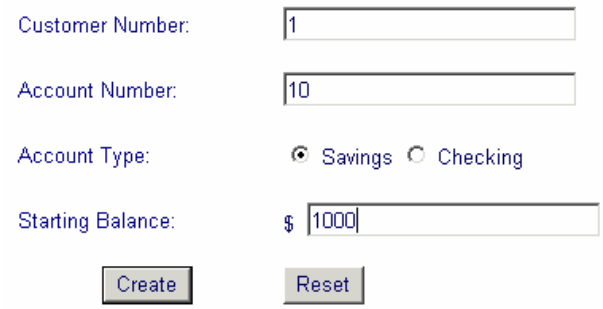
__ j. Test the WebSphereBank application. Open another web browser and type in the URL http://<HOST_NAME>:9080/WebSphereBankWeb/index.html to access the application.



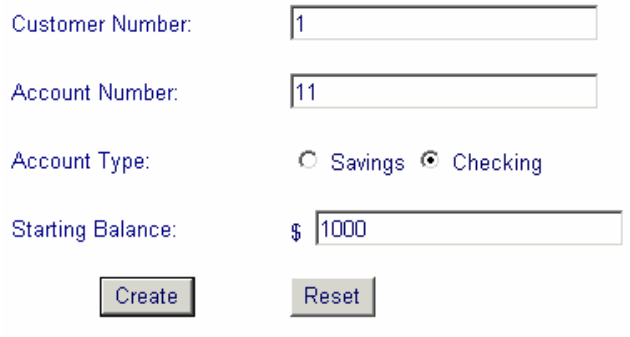
__ k. Create a customer. Click on **Create Customer**. Type in **1** for the Customer Number, **John** for the First Name, **Doe** for the Last Name, and **123456789** for the TAX ID. Click **Create.** A "New Customer has been successfully created" message will appear.



__ l. Create an account for the customer. Click on **Create Account**. For Customer Number type in **1**. For Account Number type in **10**. Choose **Savings** as the type of account. Enter in **1000** as the starting balance. Click **Create.**



__ m. Create another account for the customer. Type in **11** for the Account Number. Choose **Checking** as the Account Type. Enter in **1000** as the starting balance. Click **Create.**



__ n. Now try to Transfer Funds between the two accounts. Click on the **Transfer Funds** link. Notice that an error is displayed stating the transferfunds.jsp is not found.

JSP Processing Error

HTTP Error Code: 404

Error Message:

```
JSPG0036E: Failed to find resource /jsp/transferfunds.jsp
```

Root Cause:

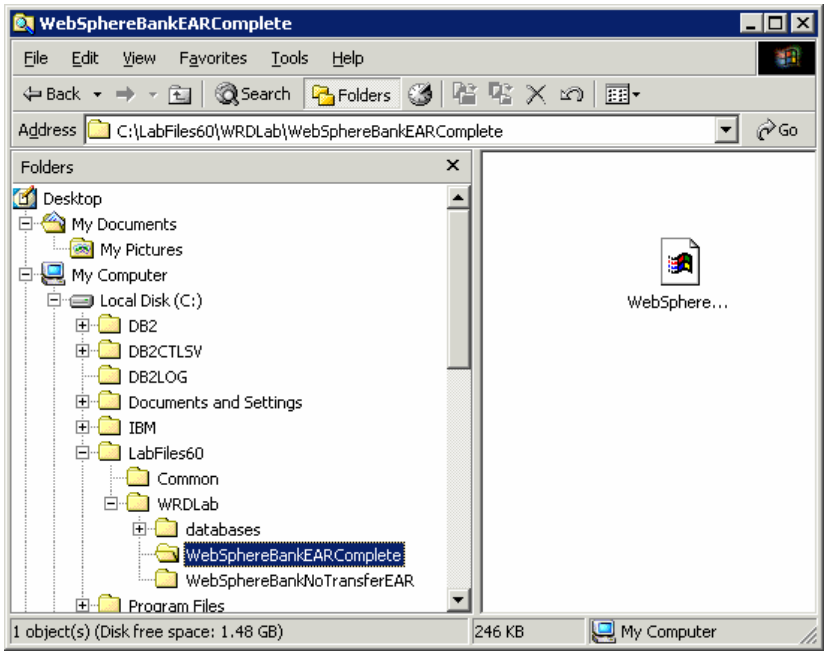
```
java.io.FileNotFoundException: JSPG0036E: Failed to find resource /jsp/transferfunds.jsp
        at com.ibm.ws.jsp.webcontainerext.JSPExtensionProcessor.findWrapper(JSPExtensionProcessor.java:244)
        at com.ibm.ws.jsp.webcontainerext.JSPExtensionProcessor.handleRequest(JSPExtensionProcessor.java:227)
        at com.ibm.ws.webcontainer.webapp.WebApp.handleRequest(WebApp.java:2752)
        at com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(WebGroup.java:220)
        at com.ibm.ws.webcontainer.VirtualHost.handleRequest(VirtualHost.java:204)
        at com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:1668)
        at com.ibm.ws.webcontainer.channel.WCChannelLink.ready(WCChannelLink.java:77)
        at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleDiscrimination(HttpInboundLink.java:380)
        at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleNewInformation(HttpInboundLink.java:341)
        at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.ready(HttpInboundLink.java:249)
        at com.ibm.ws.tcp.channel.impl.NewConnectionInitialReadCallback.complete(NewConnectionInitialReadCallback.java
        at com.ibm.ws.tcp.channel.impl.WorkQueueManager.requestComplete(WorkQueueManager.java:527)
        at com.ibm.ws.tcp.channel.impl.WorkQueueManager.attemptIO(WorkQueueManager.java:580)
        at com.ibm.ws.tcp.channel.impl.WorkQueueManager$Worker.run(WorkQueueManager.java:945)
        at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1292)
```

__ o. This WebSphereBank application that was installed does not contain the Transfer Funds functionality.

__ p. **Close** the web browser that is accessing the WebSphereBank application.

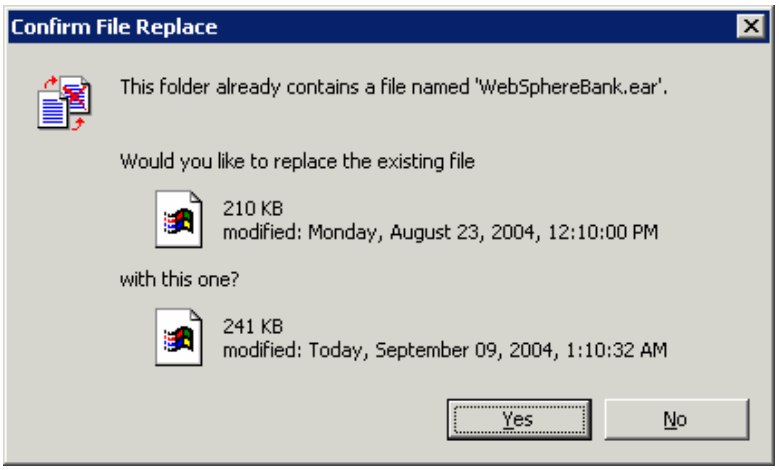
____ 6. Update the application with the Transfer Funds capability.

__ a. Open Windows Explorer and navigate to the **C:\LabFiles60\WRDLab\WebSphereBankEARComplete** directory.



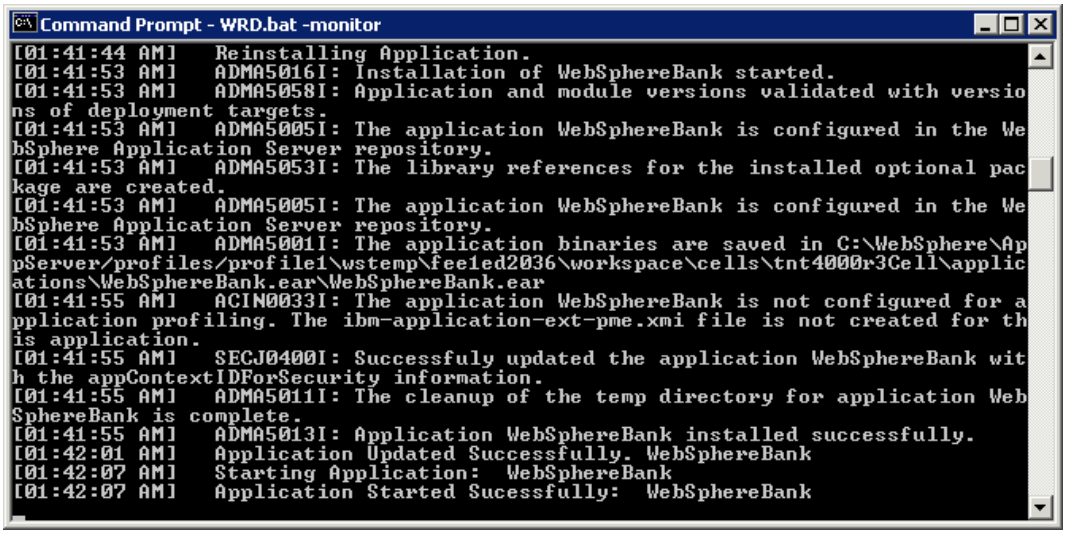
__ b. **Right-click** on the file WebSphereBank.ear and choose **Copy.**

__ c. In Windows Explorer, navigate to C:\WRD\AutoInstall and overwrite the existing WebSphereBank.ear file by clicking Yes in the Confirm File Replace dialog box.



Note: This new WebSphereBank application does have the Transfer Funds functionality.

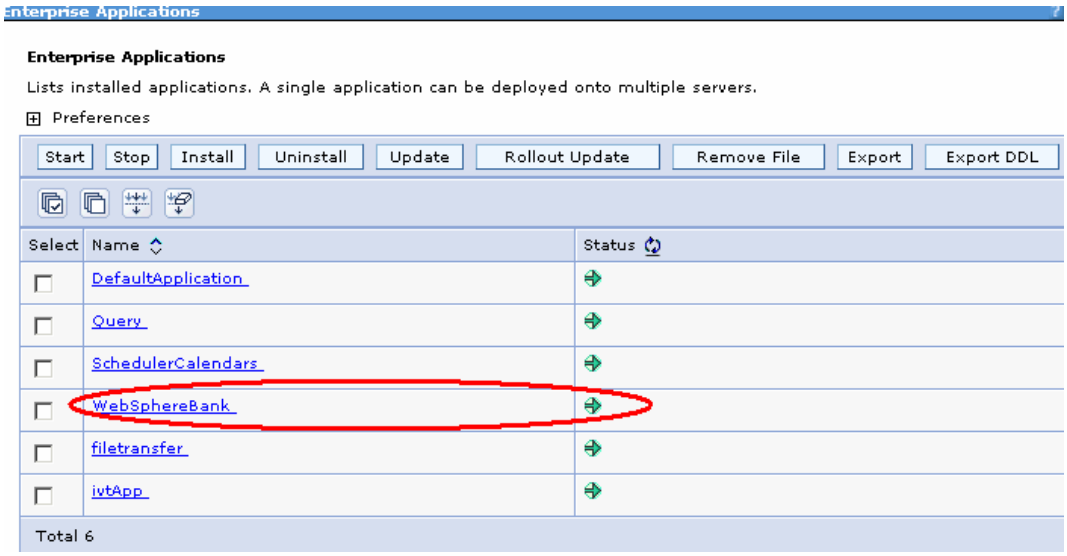
__ d. Check the command prompt where WRD was started, it should resemble the screen capture below.



Note: Notice that the application has been updated.

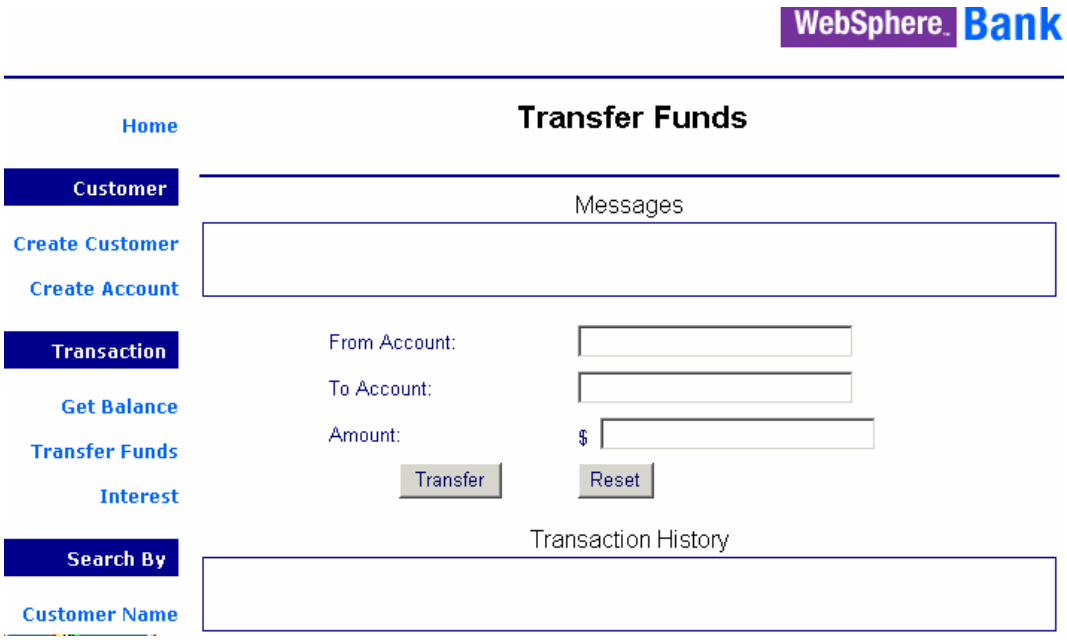
__ e. Go back to the web browser which is accessing the administrative console. Click **Logout** and then log back into the console using **wsdemo** has the User Id.

__ f. Expand Applications and then click Enterprise Applications. Verify that the application is up and running.

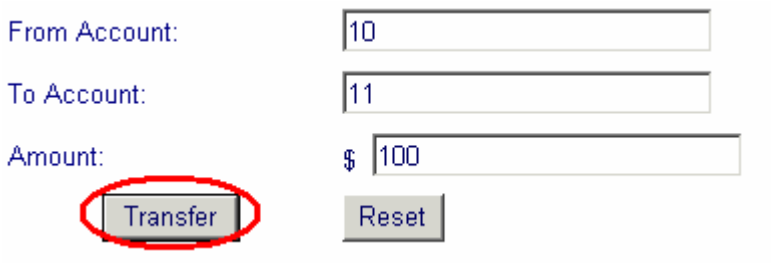


__ g. Open another web browser and access the WebSphereBank application by going to the URL: http://<HOST_NAME>:9080/WebSphereBankWeb/index.html.

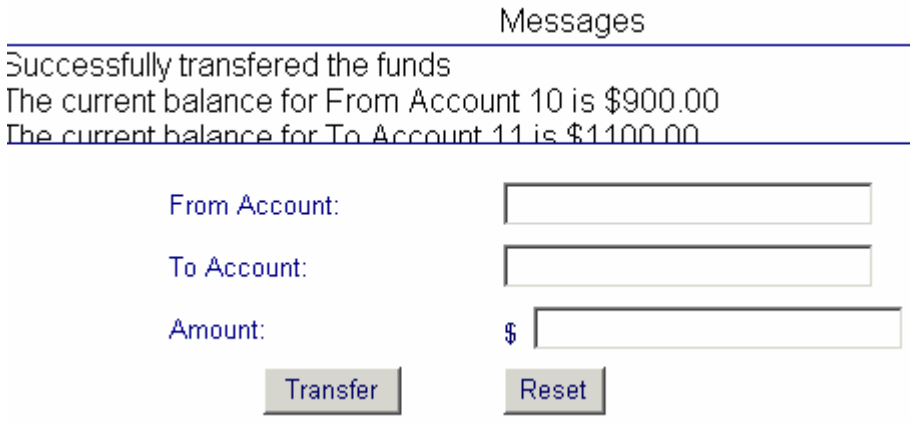
__ h. At the main page of WebSphereBank click on the **Transfer Funds** link. Notice that the jsp to perform the transfer is now present in the application.



__ i. Transfer funds between accounts. For the From Account field, type in **10**. For the To Account field, type in **11**. For the Amount, type in **100** and click on **Transfer**.



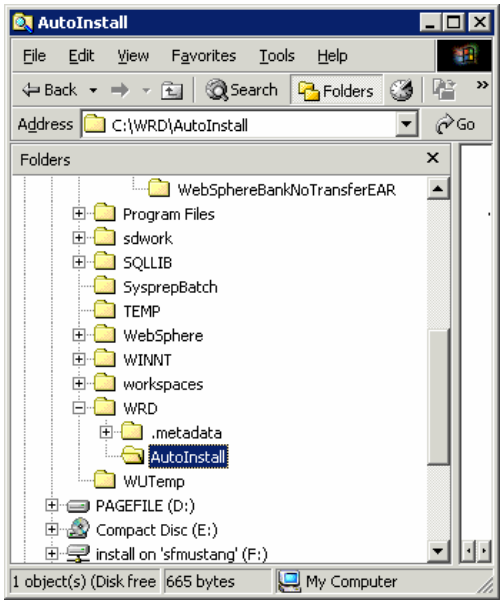
__ j. The following screen capture should appear stating the transfer was successful and show what the new balances are for each account.



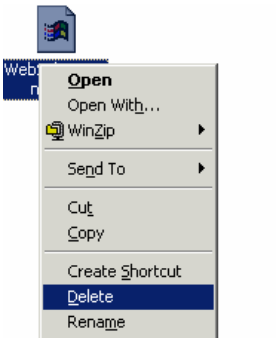
__ k. **Close** the web browser that is accessing the WebSphereBank application.

____ 7. Delete the application from the WRD Project to uninstall it from the Application Server.

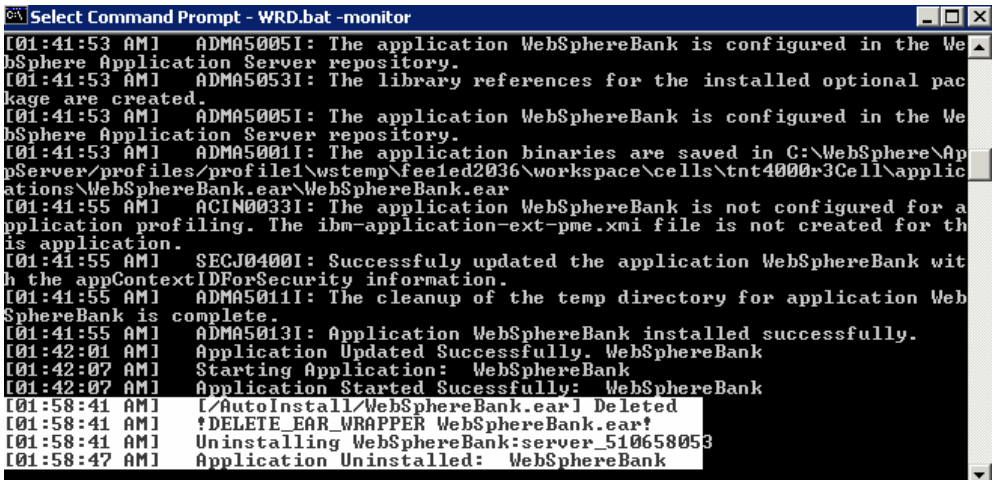
__ a. In Windows Explorer, navigate to C:\WRD\AutoInstall directory



__ b. Right-click on the file **WebSphereBank.ear** and choose **Delete**.

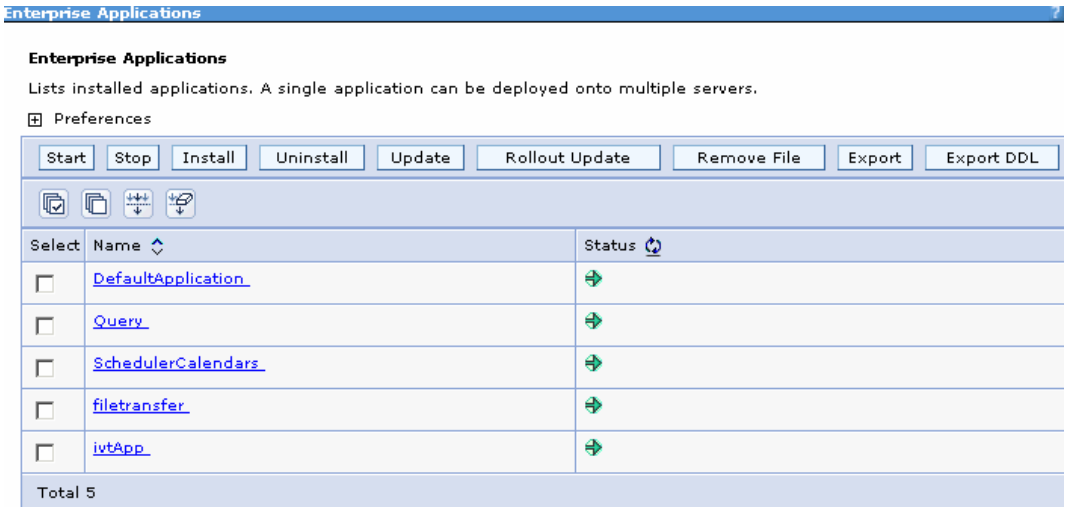


__ c. Go to the command prompt where WRD was started. After a minute the following output should appear indicating the application was uninstalled.



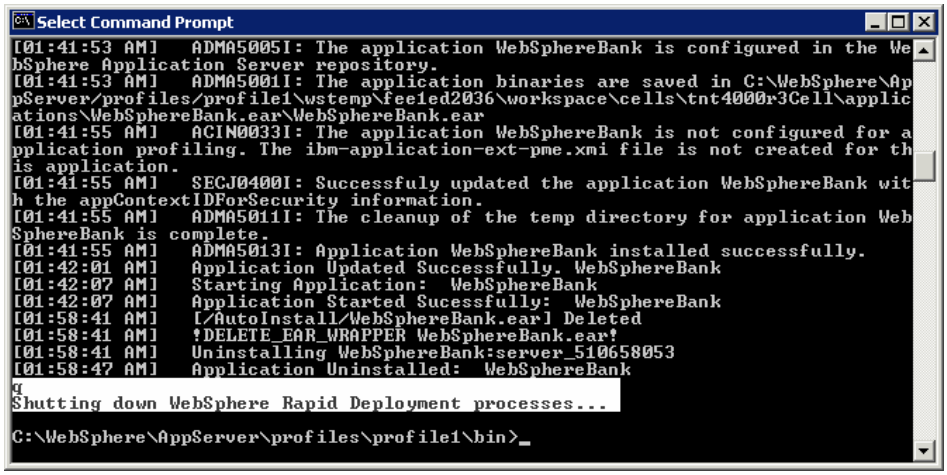
__ d. Go back to the web browser which is accessing the administrative console. Click **Logout** and then log back into the console using **wsdemo** has the User Id.

__ e. Expand Applications and then click Enterprise Applications. Verify that the application has been uninstalled.



__ f. **Log out** of the administrative console and **close** the web browser.

__ g. In the WRD command prompt, type **q,** and press **Enter**, to stop the WRD project.



__ h. In Windows Explorer, navigate to C:\WRD and delete the **AutoInstall** directory.

This completes the autoappinstall style portion of the lab.

Part 5: WRD Deployment Automation "freeform" style: simple application

____ 1. Configure WRD project to support the freeform style for a simple web application that contains a HelloWorld servlet. The *free-form* configuration allows you to create or drop in your J2EE artifacts or module files into the free-form project. J2EE artifacts includes the source, annotated-source or class files of servlets, JSP, static Web content, enterprise beans, and other generic files. The rapid deployment tools will then use your artifacts dropped in this single project directory to automatically place them in the appropriate J2EE project structure, generate any additional required artifacts to construct a J2EE compliant application and deploy that application on a target server.

__ a. At the command prompt, navigate to **C:\IRAD60\runtimes\base_v6\bin** if that is not current directory

__ b. To configure the WRD project type: **wrd-config –project "Hello" –style "freeform"** at the command prompt and press **Enter**.

__ c. This command will execute for a minute as it starts to build the project. It will then ask for configuration parameters.

1) Fill in the server name: ex. G6Server1

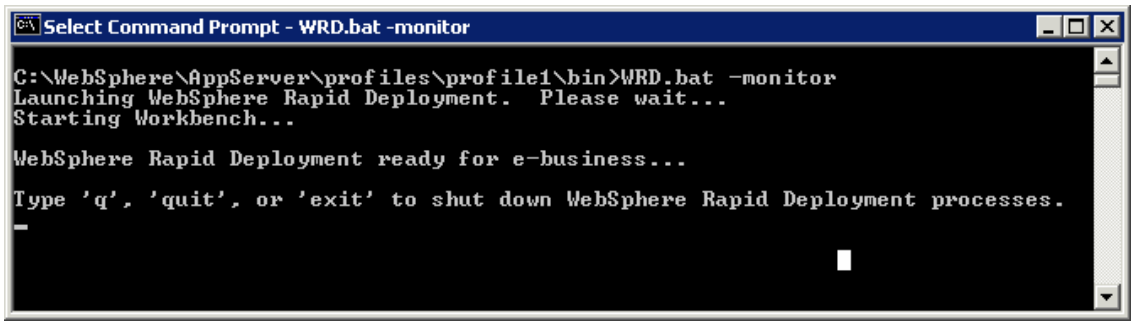
2) Fill in JMX host name: ex. mvs22x.rtp.raleigh.ibm.com

3) Fill in JMX port: ex. 8880

4) Accept the defaults for the rest of the parameters by pressing enter after each one.

____ 2. Start the WRD project.

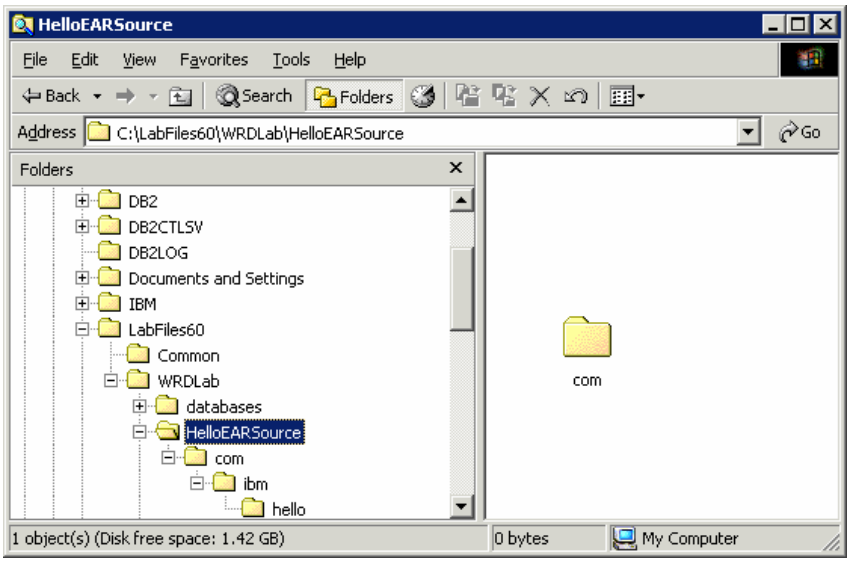
__ a. At the command prompt type: **WRD.bat –monitor** and press **Enter.** The command may take a minute to process. After it's complete the following output will appear:



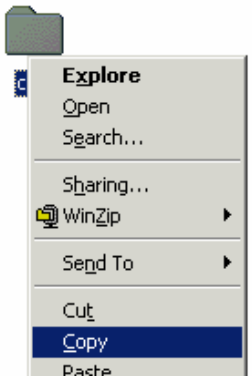
Note: Specifying the –monitor parameter will allow the console output to be directed at the command prompt window to display error, success, and overall status messaging. A log file can also be used.

____ 3. Drop a simple application's source code into the WRD project and examine the enterprise application that is constructed, deployed, and installed onto WebSphere Application Server.

__ a. Open Windows Explorer and navigate to the directory **C:\LabFiles60\WRDLab\HelloEARSource**.

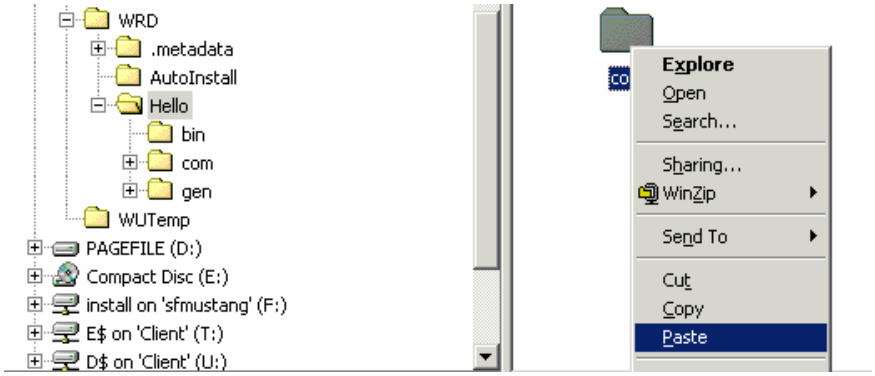


__ b. Right-click on the folder com and choose Copy.

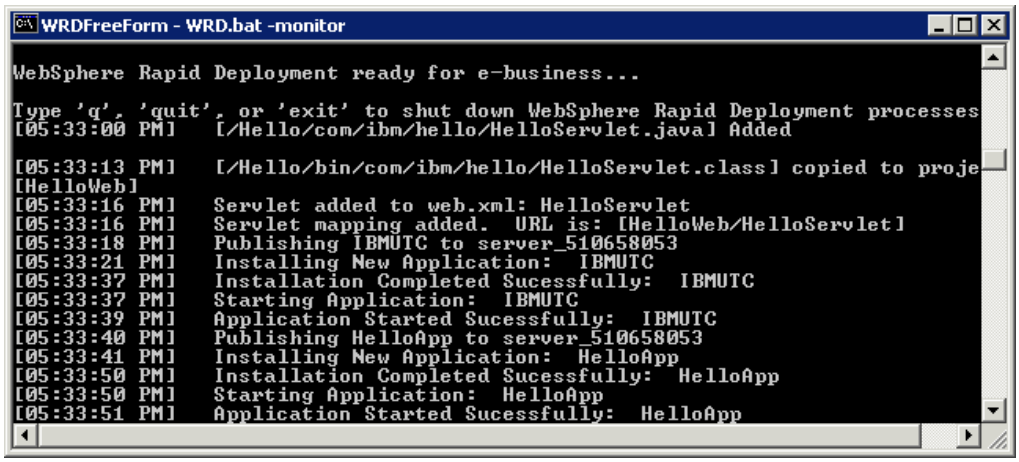


__ c. In Windows Explorer, navigate to **C:\WRD\Hello** directoy.

__ d. **Paste** the “com” directory previously copied in step b in the **C:\WRD\Hello** directory.



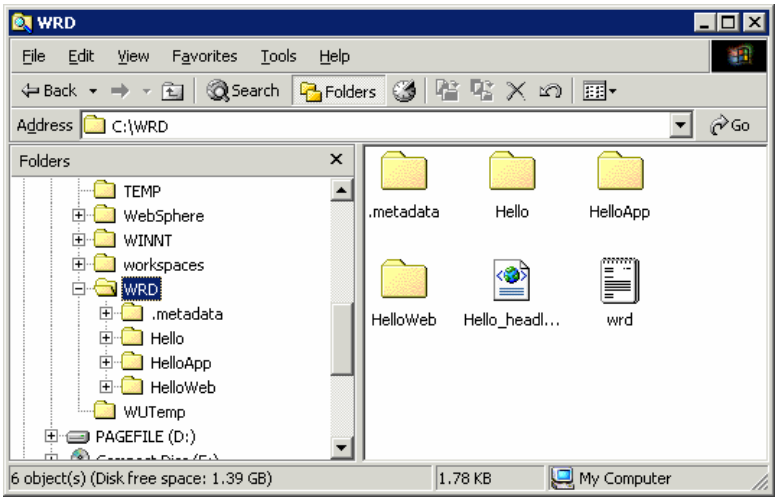
__ e. Go to the command prompt where WRD was started and the following output should appear stating the application was constructed correctly and deployed, installed and started on WebSphere Application Server.



__ f. To verify the application HelloApp is running. Open a web browser and go to http://<HOST_NAME>:9080/HelloWeb/HelloServlet. The message HELLO WORLD!!!! will appear.

__ g. **Close** the browser.

__ h. Now explore the artifacts created to package the application in a J2EE format and deploy/install on WebSphere Application Server. Open Windows Explorer and navigate to **C:\WRD.** The follow directories should appear:



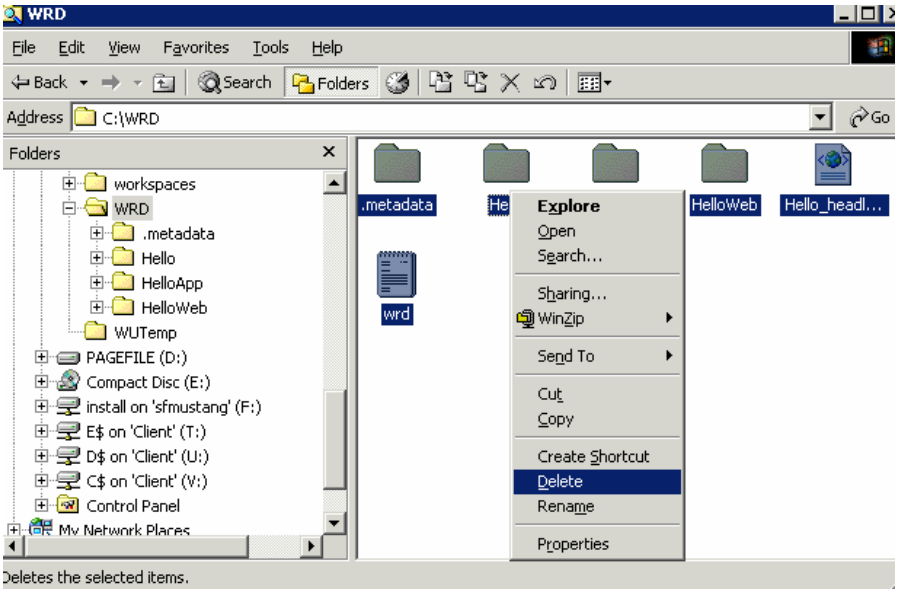
__ i. **Hello Folder**: Monitored WRD project folder.

__ j. **HelloApp Folder**: Enterprise application folder for J2EE application artifacts

__ k. **HelloWeb Folder**: Web Module project for J2EE web artifacts

__ l. In the command prompt where WRD was started, type **q** and press **Enter** to quit WRD.

__ m. In Windows Explorer, navigate to **C:\WRD** and **delete** all files and directories that reside under the WRD folder.



What you did in this exercise

In this exercise you worked with both styles of WebSphere Rapid Deployment, deployment automation. In the first part you installed/updated/uninstalled an application by simply moving/copying/removing the EAR file into a specified project directly. In the second part, you created a project using the freeform style that automatically constructed an EAR file and installed it on to WebSphere Application Server.

This page is left intentionally blank.