

## IBM WEBSHERE WORKSHOP - LAB EXERCISE

# Install an Application in WebSphere AEs V4.0

### **What This Exercise is About**

You will install an application in WebSphere Advanced Edition Single Server (AEs) Version 4.0. WebSphere AEs provides a command line installation utility called SEAppInstall, which simplifies the installation process. WebSphere V4.0 is a J2EE compliant application server, so applications must be packaged as J2EE Enterprise Archive files with a .ear extension. The Enterprise Application contains all of the application components, such as EJBs, Servlets, HTML and image files. The Application .ear file also contains deployment descriptors, which describe how to deploy in a J2EE compliant server like WebSphere. Applications created in WebSphere's Application Assembly Tool (AAT) also contain WebSphere specific information, like IBM extensions and bindings.

### **User Requirement**

This lab requires one system with Windows NT Server 4.0, ServicePack 6. IBM DB2 Universal Database Version 7.2a is required. This lab is based on IBM WebSphere Application Server, Release 4.0.1 Advanced Single Server Edition, driver AEs 4.0.1 a0131.07. This lab requires the MyBank Enterprise Archive file MyBankApp.ear.

### **What You Should Be Able to Do**

You should be able to use the SEAppInstall command line utility to install a J2EE application in WebSphere Advanced Single Server Edition (AEs) Version 4.0. You will use the EARExpander command line utility to expand the contents of an Enterprise Archive .ear file. This allows you to see all of the files contained inside the archive. You will create an application DataSource in WebSphere, and create the DB2 database for the DataSource. You will create the table in the database, create a user for database access, and grant that user the required database privileges. You will start the WebSphere application server, and test the sample application.

### **Introduction**

You will install an application called MyBank, which simulates a very simple online bank. The application allows you to create checking and savings accounts, check your account balances, and transfer funds between your accounts. The application has two interfaces. The Web interface runs in a browser. The Java client shows how a stand alone Java application can access the balance and transfer functions in the EJBs without going through an HTTP Server.

## Exercise Instructions

This lab consists of six parts:

- Part One: Examine the Application .ear file
- Part Two: Create the Application DataSource in WebSphere
- Part Three: Install the Application in WebSphere AEs
- Part Four: Create the Application Database in DB2
- Part Five: Test the Application in WebSphere AEs
- Part Six: View Deployment Descriptors and Mappings through Browser Console

### Part One: Examine the Application .ear file

- \_\_ 1. Before installing the MyBank application, take a look at the contents of the .ear file. The Enterprise Archive contains all of the application code components, deployment descriptors, IBM extensions and bindings.

- \_\_ Click the Windows **Start** button, then select: **Run**
- \_\_ In the **Open** field of the run window, type: **cmd**
- \_\_ Click **OK**.
- \_\_ At the prompt in the Command window, type: **cd %WAS\_HOME%\bin**
- \_\_ At the prompt in the Command window, type (all on one line): **EARExpander -ear C:\MyBank\MyBankApp.ear -expandDir C:\MyBankExpanded -operation expand -expansionFlags all**
- \_\_ Press the **Enter** key.

- \_\_ 2. Open Windows Explorer to see the expanded directory.



- \_\_ Click the Windows **Start** button
- \_\_ Select **Programs --> Windows NT Explorer**.
- \_\_ Click the **+** next to the **MyBankExpanded** directory on the left.
- \_\_ If you do not see the MyBankExpanded directory, press **F5** to refresh the view.
- \_\_ Note that there is a subdirectory with the same name as each of the Modules.
- \_\_ Select the **Meta-inf** directory, which contains information about the application.

- \_\_ 3. The application deployment descriptor is an XML file called application.xml.

- \_\_ Right click **application.xml** and select **Send To --> Wordpad**.
- This simple file shows that the application contains three modules:
 

EjbModule_1	Deployed_MyBankEJB.jar
WebModule_1	MyBankWeb.war
JavaClientModule_1	MyBankClient.jar

This file also shows that the Web Module context-root is /MyBank.

The context root is part of the URL to all of the application components.  
The application URL for MyBank is: <http://hostname:port/MyBank>.

- \_\_4. The EJB Module deployment descriptor is an XML file called `ejb-jar.xml`.

\_\_ In Windows Explorer, click the **+** next to the **Deployed\_MyBankEJB.jar** directory.

\_\_ Select the **Meta-inf** directory, which contains information about the EJB Module.

\_\_ Right click **ejb-jar.xml** and select **Send To --> Wordpad**.

This file describes the two Enterprise JavaBeans contained in the Module:

ContainerManagedEntity_1	Account
Session_1	Transfer

This file also shows an assembly descriptor which requires a transaction context for the Account and Transfer beans. If you want to transfer funds between accounts, both account balances need to change. If the system cannot change the balance of one of the accounts for any reason, the entire transaction should roll back.

- \_\_5. The Web Module deployment descriptor is an XML file called `web.xml`.

\_\_ In Windows Explorer, click the **+** next to the **MyBankWeb.war** directory.

\_\_ Select the **Web-inf** directory, which contains information about the Web Module.

\_\_ Right click **web.xml** and select **Send To --> Wordpad**.

This file describes the three Servlets contained in the Module:

Servlet_1	CreateAccount
Servlet_2	TransferFunds
Servlet_3	ErrorReporter

The Servlet mapping entries show the URL pattern associated with each Servlet:

<u>servlet-mapping_id</u>	<u>servlet-name</u>	<u>url-pattern</u>
ServletMapping_1	CreateAccount	servlet/Create
ServletMapping_2	TransferFunds	servlet/Transfer
ServletMapping_3	ErrorReporter	servlet/Error

The URL pattern follows the context root when calling the Servlet from a browser:

The URL for CreateAccount is: <http://hostname:port/MyBank/servlet/Create>

The URL for TransferFunds is: <http://hostname:port/MyBank/servlet/Transfer>

- \_\_6. The Application Client Module deployment descriptor is called `application-client.xml`.

\_\_ In Windows Explorer, click the **+** next to the **MyBankClient.jar** directory.

\_\_ Select the **Meta-inf** directory, which contains information about the Client Module.

\_\_ Right click **application-client.xml** and select **Send To --> Wordpad**.

This file describes the EJB Reference contained in the Module:

EjbRef_1	TransferLocal
----------	---------------

The Application Client calls the EJBs without going through an HTTP Server or Servlets. Some additional information about the Application Client is in the **Manifest.mf** file.

Main-Class: WebSphereSamples.AccountAndTransfer.TransferApplication  
The Main class implements the Client stand alone Java application.

Class-Path: Deployed\_MyBankEJB.jar ./  
The Class path indicates the EJBs called by the Application Client.

## **Part Two: Create the Application DataSource in WebSphere**

- \_\_1. Now that you understand what the application contains, you can prepare the WebSphere environment for the application. A DataSource is needed to configure the connection to the database where application data is stored. The application stores information about the default DataSource in the **ibm-ejb-jar-bnd.xmi** file in the **Deployed\_MyBankEJB.jar/Meta-Inf** directory.

```
jndiName="jdbc/MyBank"  
userId="USERID" password="{xor}Dx4MDAgQDRs="
```

Note that the password is not stored in clear text. The password is not actually encrypted, just encoded.

- \_\_2. Start WebSphere Advanced Edition Single Server V4.0.

\_\_ At the prompt in the Command window, type: **cd %WAS\_HOME%\bin**

\_\_ At the prompt in the Command window, type: **startserver**

\_\_ The message **Server Default Server open for e-business** indicates startup is finished.

- \_\_3. Start the WebSphere Advanced Edition Single Server V4.0 console in a browser.

\_\_ Open a browser window, either Netscape Navigator or Internet Explorer.

\_\_ Open the URL: <http://localhost:9090/admin>

\_\_ In the Login window, fill in User ID: **C3PO** (or any value), and click **Submit**.

\_\_ On the left, click the **+** next to **Resources**.

\_\_ Click the **+** next to **JDBC Drivers**.

\_\_ Select **Db2JdbcDriver**. The properties appear on the right.

The problem is that the Server Class Path is empty. The path to the DB2 JDBC driver class `COM.ibm.db2.jdbc.DB2ConnectionPoolDataSource` is needed.

To find where DB2 is installed on your system, in a command window type: **set db2**. This displays all the environment variables that start with db2. The variable DB2TEMPDIR shows where DB2 is installed. The value is probably either C:\SQLLIB or C:\PROGRA~1\SQLLIB.

- \_\_4. Set the JDBC Driver Class Path to the correct value for your system.
- \_\_ Fill in the Server Class Path: **C:/SQLLIB/java/db2java.zip**  
or **C:/Program Files/SQLLIB java/db2java.zip**
  - Note the use of forward slashes / in the Path.
  - \_\_ Click **OK**.
  - \_\_ Click the Information message: **Configuration needs to be saved**.
  - \_\_ In the Save Configuration window, click **OK**.
  - \_\_ This will Save Configuration File: server-cfg.xml, the Administrative Repository.
- \_\_5. Create the DataSource for the MyBank application.
- \_\_ On the left, click the **+** next to **Db2JdbcDriver**.
  - \_\_ Click to select **DataSources**.
  - \_\_ On the right, click **New**.
  - \_\_ In the New Data Source window, fill in:

Name:	<b>BankData DataSource</b>
JNDI Name:	<b>jdbc/MyBank</b>
Description:	<b>DB2 database for MyBank application</b>
Category:	<b>MyBank</b>
Database Name:	<b>BankData</b>
Default User ID:	<b>USERID</b>
Default Password:	<b>PASSWORD</b>
Minimum Pool Size:	<b>1</b>
Maximum Pool Size:	<b>10</b>
Connection Timeout:	<b>1000</b>
Idle Timeout:	<b>2000</b>
Orphan Timeout	<b>3000</b>
Statement Cache Size:	<b>100</b>
Disable Auto Connection Cleanup:	(Do not check this box)

\_\_ Click **OK**.

The screenshot shows the IBM WebSphere Application Server Administrative Console. The browser window title is "IBM WebSphere Application Server Administrative Console - Microsoft Internet Explorer". The address bar contains "http://localhost:9090/admin/secure/logon.do?action=OK". The console header displays "WebSphere Application Server" and "server-cfg.xml". A black arrow points to the "Save" button in the top navigation bar. A message "Configuration needs to be saved" is displayed. Below it, the "Data Sources" section is visible, containing a table with two entries:

<input type="checkbox"/>	Name	JNDI Name	Description	C
<input type="checkbox"/>	<a href="#">Session Persistence datasource</a>	jdbc/Session	Data source for session persistence	S:
<input type="checkbox"/>	<a href="#">MyBank DataSource</a>	jdbc/MyBank	DB2 DataSource for MyBank Application	M

\_\_ A message indicates that Configuration needs to be saved.

\_\_ Click **Save** in the black menu bar.

\_\_ In the Save Configuration window, click **OK**.

This will Save Configuration File: server-cfg.xml, the Administrative Repository.

\_\_6. Remove an Enterprise Application which conflicts with MyBank. WebSphere AEs comes with several applications installed, and one has a naming conflict with MyBank.

\_\_ In the Console window, click the **+** next to **Nodes**.

\_\_ Click the **+** next to the **node** name. This is the short host name of your machine.

\_\_ Click to select **Enterprise Applications**.

- \_\_ On the right, click the box  next to **WebSphere Application Server Samples** to select it.
- \_\_ Click **stop** in the Enterprise Application window.
- \_\_ Click **Uninstall** in the Enterprise Applications window.
- \_\_ Click **Uninstall** in the Specify the Application that you want to Uninstall window.
- \_\_ Click **Save** in the black menu bar.
- \_\_ In the Save Configuration window, click **OK**.

\_\_7. The WebSphere server is now ready for installation of MyBank application.

- \_\_ In the command window, type **stopserver** to stop the server.

### **Part Three: Install the Application in WebSphere AEs**

\_\_1. Installing an application in WebSphere AEs is remarkably simple. A single command line utility called SEAppInstall does the entire job for you. The utility generates many messages in the command window, so resize the Command window to fill the screen.

- \_\_ Right click the menu bar in the Command window, and select **Properties**.
- \_\_ In the Properties window, click the **Layout** tab, and fill in:

Screen Buffer Size:	Width:	<b>125</b>
	Height:	<b>500</b>
Window Size:	Width:	<b>125</b>
	Height:	<b>55</b>

- \_\_ Click **OK**. Click **OK**. Drag the menu bar to center the Command window.
- \_\_ At the prompt in the Command window, type: **cd %WAS\_HOME%\bin**
- \_\_ At the prompt in the Command window, type (all on one line):

**SEAppInstall -install C:\MyBank\MyBankApp.ear**

- \_\_ Respond **No** to the following questions in the Command window:

```
Do you wish to deploy all of the EJBs in this application ([Y]es/[n]o)? N
The EJB Deploy Option has been disabled. Deployment skipped.
Do you wish to precompile all JSPs in this application ([Y]es/[n]o)? N
Do you wish to precompile individual Web Applications ([Y]es/[n]o)? N
```

- \_\_ Press **Enter** to accept the values from the .ear file for the rest of the questions:

Please provide the following EJB Jar Binding Information

```
-----
EJB Jar: Deployed_MyBankEJB.jar
-----
Default Datasource JNDI Name (optional) [jdbc/MyBank]:
```

User ID [USERID]:  
Password [PASSWORD]:

Enterprise Bean: WebSphereSamples.AccountAndTransfer.AccountBean  
-----

JNDI Name [AccountHomeGlobal]:  
Datasource JNDI Name []:

Enterprise Bean: WebSphereSamples.AccountAndTransfer.TransferBean  
-----

JNDI Name [TransferHomeGlobal]:

No Resource References Defined as part of the Application

The following EJBs are available on this node.

Format: n) <Bean Name> [<JNDI Name>]

- 
- 1) WebSphereSamples.AccountAndTransfer.AccountBean [AccountHomeGlobal]
  - 2) WebSphereSamples.AccountAndTransfer.TransferBean [TransferHomeGlobal]

Please Specify a JNDI Name for the following EJB References

- 
- 1) AccountLocal (MyBank Web Module) [AccountHomeGlobal]:
  - 2) TransferLocal (MyBank Web Module) [TransferHomeGlobal]:
  - 3) AccountLocal (MyBank EJB  
Module:WebSphereSamples.AccountAndTransfer.TransferBean) [AccountHomeGlobal]:

No Security Roles Defined as Part of Application

No "Run As Roles" Mappings defined

Please specify a Virtual Host for the following Web Applications

- 
- 1) MyBank Web Module (WebApp\_1) [default\_host]:

If your screen does not match these default values, then there is an error in your MyBankApp.ear file. You can supply the correct values during the installation process.

SEAppInstall installs the application .ear file in the WebSphere Application Server.

A new directory named **MyBankApp.ear** is created under the **installedApps** directory.

The **MyBankApp.ear** directory is the partially expanded version of the .ear file.

#### **Part Four: Create the Application Database in DB2**

- \_\_1. When you created the DataSource, you specified that the Default User ID for the application database is **USERID**, and the Default Password is **PASSWORD**. You need to create this User ID in the operating system registry.

\_\_ Click the Windows **Start** button, and select **Programs --> Administrative Tools (Common) --> User Manager for Domains**. (On NT Workstation, **User Manager**)

\_\_ In the User Manager window, click to select Username **Guest**.

\_\_ Click the **User** menu, and select **Copy**. In the Copy of Guest window, fill in:

Username: **USERID**  
 Full Name: **WebSphere Lab**  
 Description: **DB2 User for MyBank Application**  
 Password: **PASSWORD**  
 Confirm Password: **PASSWORD**

\_\_ Click **Add**. Click **Close**. (or **Cancel** on NT Workstation.) Exit the **User Manager**.

- \_\_2. Before starting WebSphere to test the application, you need to create the application database in DB2. When you created the DataSource, you specified that the database is named BankData. During code generation, WebSphere created a *Table.ddl* file with an SQL description of the database table in the *Meta-Inf* directory of the EJB Module.

\_\_ In a Command window, change to the directory for **Deployed\_MyBankEJB.jar**.  
 For example: **cd C:\WebSphere\SingleServer\installedApps\MyBankApp.ear**  
 \_\_ Extract the Table.ddl file from the MyBankEJB.jar file with this command:

```
jar -xf Deployed_MyBankEJB.jar META-INF\Table.ddl
```

\_\_ If you get an error message: *The name specified is not recognized as an internal or external command, operable program or batch file*, then the WebSphere java directory needs to be added to your system's Path environment variable:

```
set Path=%Path%;%WAS_HOME%\java\bin
```

An alternative method to extract the Table.ddl file is to use the Web browser console. Start the WebSphere Application server with the command **startserver**.

Click **+** to expand **Nodes** and **Node Name**, then select **Enterprise Applications**. You can select an application, click the **Export DDL** button, and download the file. This application will download a file named **MyBankApp\_Deployed\_MyBankEJB.ddl**. In the Save dialog box, you can rename the file to **Table2.ddl**, and save it in the **C:\WebSphere\SingleServer\installedApps\MyBankApp.ear\META-INF** directory.

\_\_3. Create the application database in DB2 using the description in Table.ddl.

\_\_ Launch a DB2 Command window with this command: **db2cmd**

\_\_ In the DB2 CLP window, type the following commands:

**db2 create db BankData**

This creates a new empty database, and takes a couple minutes to complete.

**db2 update db cfg for BankData using applheapsz 256**

**db2 connect to BankData**

**db2 grant createtab, connect on database to user USERID**

This grants basic database privileges to the User you just created.

**db2 connect reset**

**db2 connect to BankData user USERID using PASSWORD**

This allows you to create the database tables under the schema for USERID.

**db2 -t -f META-INF\Table.ddl**

**db2 list tables**

This shows a single table named ACCOUNT was created.

**db2 select \* from ACCOUNT**

This shows the ACCOUNT table has fields called accountId, balance, and type.

**cd c:\sqllib\bnd**

**db2 bind @db2cli.lst blocking all grant public**

**db2 connect reset**

\_\_ Close the DB2 CLP window.

\_\_4. When IBM DB2 Universal Database Version 7 is first installed, it uses the JDBC 1.0 API for connections. WebSphere V4.0 and J2EE require JDBC 2.0. Stop the DB2 JDBC Applet Server, then run a batch file to change from JDBC 1.0 to JDBC 2.0.

\_\_ Click the Windows **Start** button, and select **Settings --> Control Panel**.

\_\_ In the Control Panel, double-click **Services**.

\_\_ In the Services window, select the **DB2 JDBC Applet Server**, and click **Stop**.

\_\_ Click **Yes**, you are sure you want to stop the DB2 JDBC Applet Server service.

\_\_ Verify the WebSphere Server is stopped. In a Command window, type: **stopserver**.

\_\_ In a Command window, change directories to where **DB2** is installed.

For example: **cd C:\SQLLIB** (Could be C:\Program Files\SQLLIB on your system.)

\_\_ In the Command window, type: **cd java12**

\_\_ In the Command window, type: **usejdbc2**

- Six files are copied, and a file named inuse is updated.
- \_\_\_ In the Command window, type: **type inuse**.
- \_\_\_ This should display: `JDBC 2.0`
- \_\_\_ In the Control Panel, double-click **Services**.
- \_\_\_ In the Services window, select the **DB2 JDBC Applet Server**, and click **Start**.

### **Part Five: Test the Application in WebSphere AEs**

- \_\_\_ 1. The application is installed, and the application environment is ready to run. Start the WebSphere application server, and test the application.

- \_\_\_ At the prompt in the Command window, type: **cd %WAS\_HOME%\bin**
- \_\_\_ At the prompt in the Command window, type: **startserver**
- \_\_\_ The message **...server open for e-business** indicates startup is finished.

- \_\_\_ 2. Access the MyBank application through the web interface.

- \_\_\_ Open a browser window, either Netscape Navigator or Internet Explorer.
- \_\_\_ Open the URL: <http://localhost:9080/MyBank/index.html>
- \_\_\_ This is the home page for the MyBank application.

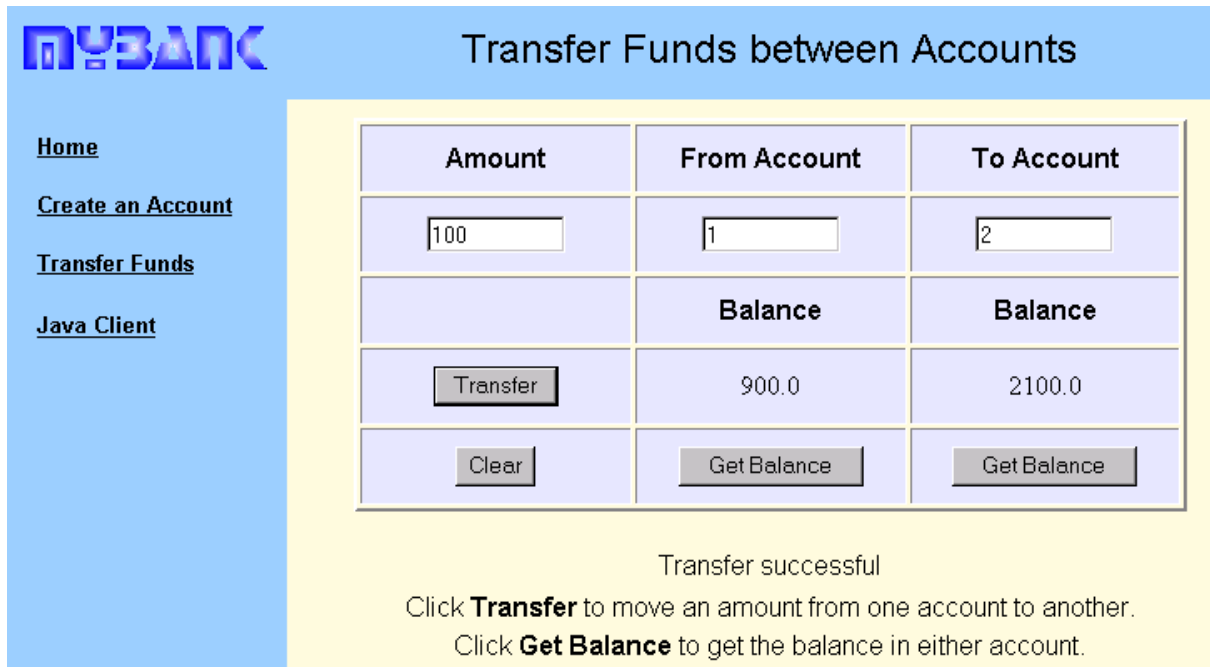
- \_\_\_ 3. In the MyBank home page, click the link to **Create an Account**.

- \_\_\_ Create a New Account:  
Account Number: **1**      Type: **Checking**      Starting Balance: **1000**
- \_\_\_ Click **Create**.
- \_\_\_ Confirmation message: *Created account: 1, Type: checking, Balance: 1000*

- \_\_\_ Create another New Account:  
Account Number: **2**      Type: **Savings**      Starting Balance: **2000**
- \_\_\_ Click **Create**.
- \_\_\_ Confirmation message: *Created account: 2, Type: savings, Balance: 2000*

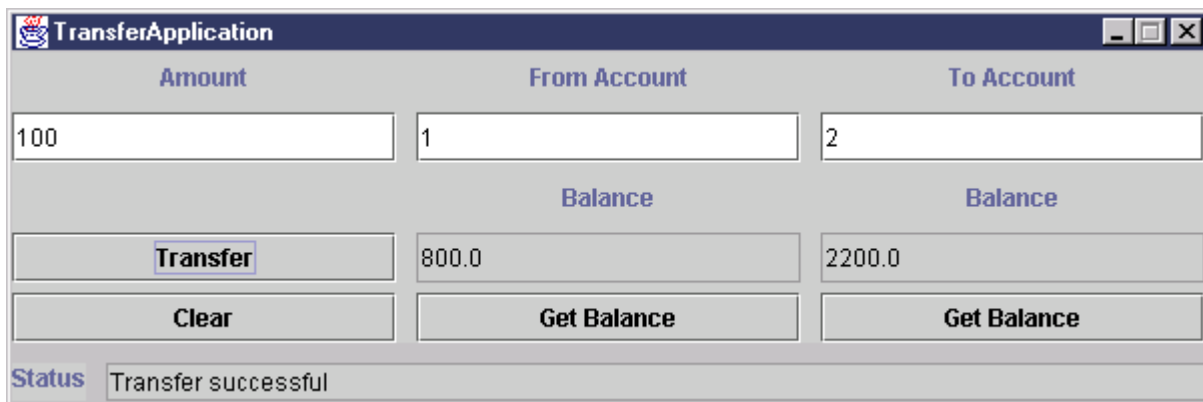
- \_\_\_ 4. In the MyBank - Create a New Account page, click the link to **Transfer Funds**.

- \_\_\_ Fill in the form to Transfer Funds between Accounts:  
Amount: **100**      From Account: **1**      To Account: **2**
- \_\_\_ Click **Transfer**.
- \_\_\_ Confirmation message: *Transfer successful*.



- \_\_5. In the MyBank - Transfer Funds between Accounts page, click the **Java Client** link. Follow the instructions on the page to launch the Java Application Client. WebSphere provides the launchClient utility to simplify the process of starting the Client.

- \_\_ Click the Windows **Start** button, select **Run**
- \_\_ On the Run window, in the Open field, type: **cmd**, click **OK**
- \_\_ In the Command window, type: **cd %WAS\_HOME%\bin**
- \_\_ At the prompt, type: **launchClient C:\MyBank\MyBankApp.ear**
- \_\_ The Java application client will start in a new window
- \_\_ Wait until the status line at the bottom says: *Initialization done - begin work.*
- \_\_ Fill in the form to Transfer Funds between Accounts:
- \_\_ Amount: **100**                      From Account: **1**                      To Account: **2**
- \_\_ Click **Transfer**.    Status: *Transfer Successful.*



## Part Six: View Deployment Descriptors and Mappings through Browser Console

1. In part one of this lab, you looked at the XML deployment descriptors for the application, EJB Module, Web Module, and Client Application. You used the EARExpander utility to expand the MyBankApp.ear file into directories, and then opened the deployment descriptors with WordPad. WebSphere V4.0 Advanced Single Server Edition also lets you view the deployment descriptors through the Console.
  - Open a browser window, either Netscape Navigator or Internet Explorer.
  - Open the URL: <http://localhost:9090/admin>
  - In the Login window, fill in User ID: **C3PO** (or any value), and click **Submit**.
  - On the left, click the **+** next to Nodes and the **+** next to the name of your Node.
  - Click the **+** next to **Enterprise Applications**.
  - Click **MyBank Application**.
  - On the right, click **View Deployment Descriptor (application.xml)**
  - Information from the XML file is displayed in a table in a new browser window.

**Application.xml**

Application: MyBank Application	
description	MyBank Enterprise Application

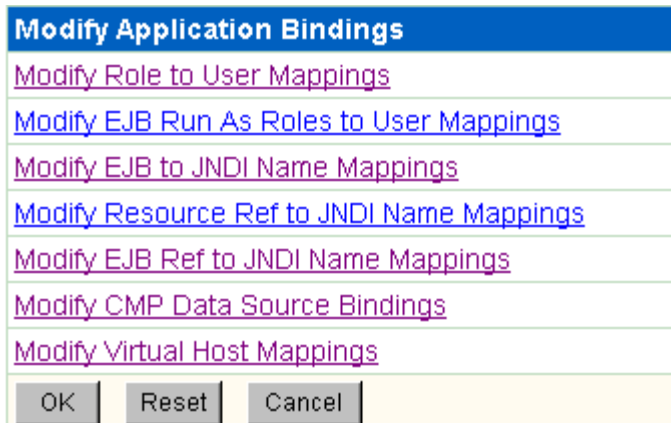
**Modules**

EJB Module	
Name	Deployed_MyBankEJB.jar

Web Module	
web-uri	MyBankWeb.war
context-root	/MyBank

App Client Module	
Name	MyBankClient.jar

From the same Console screen, you can also Modify Application Bindings:



- \_\_2. View the EJB Module deployment descriptor from the Console.
  - \_\_ On the left, click the **+** next to **MyBank Application** to expand it.
  - \_\_ On the left, click **EJB Modules**.
  - \_\_ On the right, click **Deployed\_MyBankEJB.jar**.
  - \_\_ Click **View Deployment Descriptor (ejb-jar.xml)**
    - Information from the XML file is displayed in a table in a new browser window.
  
- \_\_3. View the Web Module deployment descriptor from the Console.
  - \_\_ On the left, click **Web Modules**.
  - \_\_ On the right, click **MyBankWeb.war**.
  - \_\_ Click **View Deployment Descriptor (web.xml)**

Information from the XML file is displayed in a table in a new browser window.

**Web.xml**

Web-App: MyBank Web Module	
description	Contains Servlets, HTML and images

**Servlets**

Servlet: CreateAccount	
description	Create checking or savings account
Class	WebSphereSamples.AccountAndTransfer.CreateAccount

Servlet: TransferFunds	
description	Transfer funds between accounts
Class	WebSphereSamples.AccountAndTransfer.TransferFunds

Servlet: ErrorReporter	
description	Shows stack trace for errors
Class	com.ibm.servlet.engine.webapp.DefaultErrorReporter

**Servlet-Mappings**

Servlet: CreateAccount	
URL Pattern:	servlet/Create

Servlet: TransferFunds	
URL Pattern:	servlet/Transfer

Servlet: ErrorReporter	
URL Pattern:	servlet/Error

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

You examined the contents of the MyBankApp.ear file by expanding the archive with the EARExpander utility. You opened the deployment descriptor XML files for the application, and for the three modules it contains: EJB Module, Web Module, and Application Client Module. You started the WebSphere Advanced Single Server Edition (AEs) Version 4.0 console in a browser, and created a DataSource for the application. You learned how to uninstall an application. You installed the MyBank application using the SEAppInstall command line utility. You created a new User ID for database access. You used DB2 commands to create the application database and table, and grant access to the new User ID. You started the WebSphere server with the application installed, and tested both the Web interface in a browser, and the stand alone Java Application Client. You viewed the deployment descriptor XML files through the Console.