

## IBM WEBSHERE 5.0 Class - LAB EXERCISE

# Installing a J2EE 1.3 Application into WebSphere

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

Administration and configuration of your WebSphere v5.0.1 server can be performed using a Web-based Administrative console, command-line interface, or through a JMX interface. With the browser-based Administrative console, WebSphere can be configured through an easy, graphical interface. Resources can be defined and applications can be installed to a WebSphere server all from the Administrative console.

### **User Requirements**

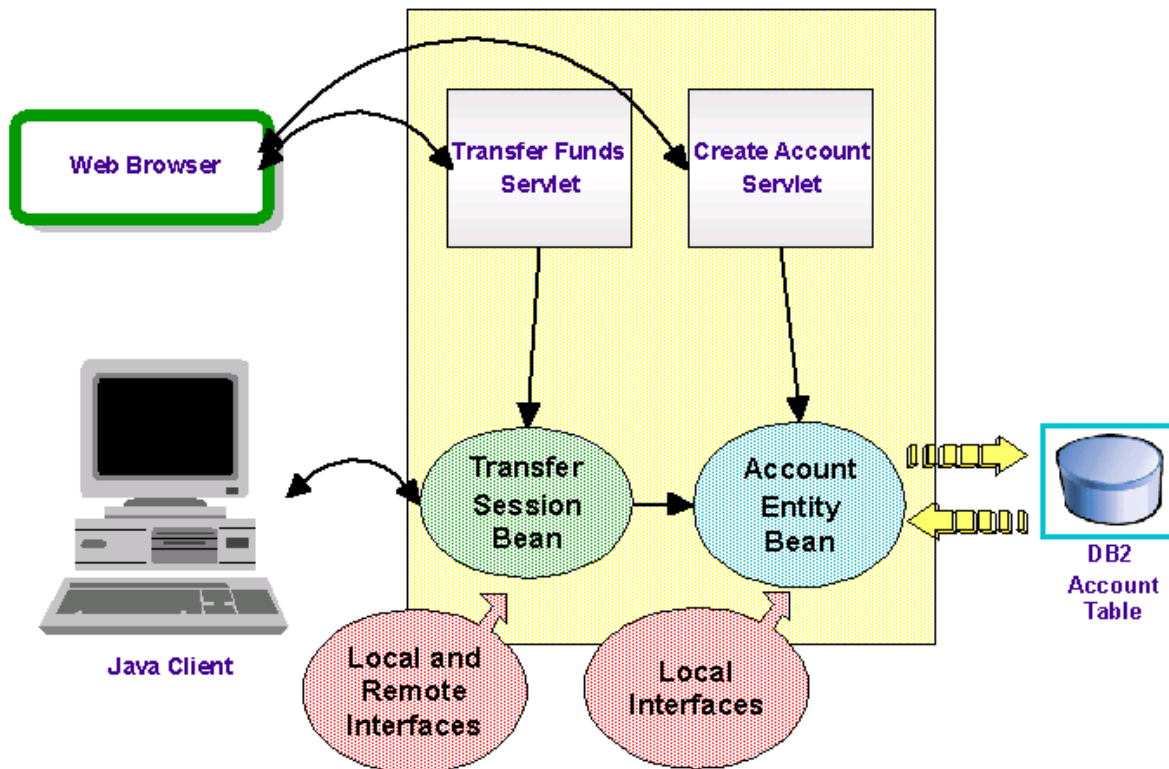
Windows 2000 Professional Service Pack 3 is required for this lab exercise. Also required is IBM WebSphere Application Server v5.0.1 and IBM DB2 Universal Database v7.2 fix pack 6. The lab source files (LabFiles50.zip) must be extracted to the root directory (i.e., C:\). Experience with previous versions of WebSphere Studio Application Developer and the J2EE programming model are also required.

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

At completion of this lab exercise, you should be able to start and stop your WebSphere server. Using the Administrative console you should also be able to define a JDBC Provider and a subsequent data source using the Administrative console. Finally, you should be able to install an application using the Administrative console and test the application.

### **Introduction**

You will begin this exercise by starting your WebSphere server. After the server has started, you will configure a JDBC Provider and a data source. The data source will be used by a J2EE application which you will install using the Administrative console. During installation you can specify binding information for the application running on your WebSphere Server. The application which you will install is a small banking application. The application features Container-Managed Persistence (CMP) and Local Interfaces, both which are part of the J2EE 1.3 specification. After you have finished installation, you will test the application through a Web interface which uses the local interfaces to create accounts and transfer funds.



**\*\* NOTE \*\*** Solution instructions are not included for this exercise as it is simply a procedural process .

## Exercise Instructions

### Part One: Configure WebSphere with Data Source

In WebSphere Application Server v5.0, the persistence model has been redesigned to support the EJB 2.0 abstract persistent schema as well as offer the ability to use persistent datastores through means besides JDBC. You will need to start your server before configuring a JDBC Provider and data source for your application.

\_\_1. Start the WebSphere Application Server.

\_\_ a. Open a command prompt and change to the WebSphere **bin** directory.

```
cd \WebSphere\AppServer\bin
```

- \_\_ b. Check on the status of the server:

```
serverstatus server1
```

- \_\_ c. If you get a message that indicates that the server appears to be stopped, enter the command to start the WebSphere server - otherwise skip this step.

```
C:\WebSphere\AppServer\bin>startserver server1
```

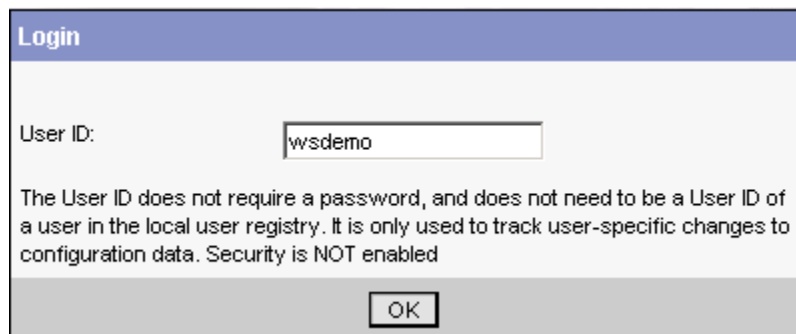
- \_\_ d. The server will be fully started when the process id is listed. The process ID can be used to track whether the server is active by viewing the running processes in the Task manager.

```
C:\WebSphere\AppServer\bin\startserver server1
ADMU0116I: Tool information is being logged in file
           C:\WebSphere\AppServer\logs\server1\startServer.log
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 2588
```

**Note:** Your process ID may be different from the one listed above

- \_\_2. The MyBank application, which you will be installing into WebSphere shortly, has an entity bean which represents an Account object. WebSphere will actually handle the persisting of the account information to a database. You will need to configure a JDBC Provider and a Data Source for the MyBank application to use for persisting data.

- \_\_ a. With the server started, open a Web browser and enter the URL <http://localhost:9090/admin>. This will start the Administrative console for administrating your WebSphere Server.
- \_\_ b. The console will open and there will be a User ID field. WebSphere allows for multiple people to administer a server having different levels of administrative rights. For your server, this feature has not been enabled and you can enter any username and click **OK**.



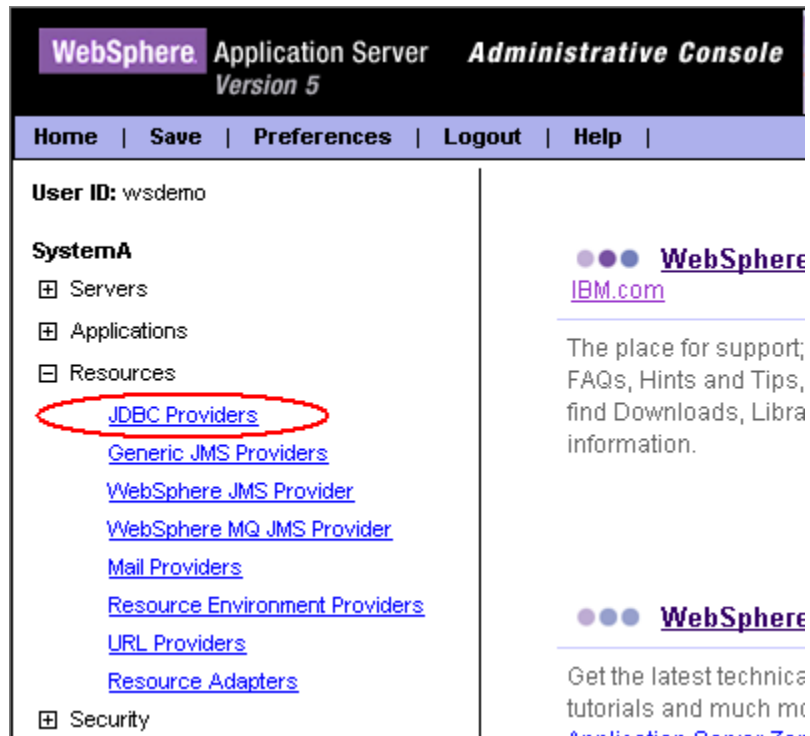
Login

User ID:

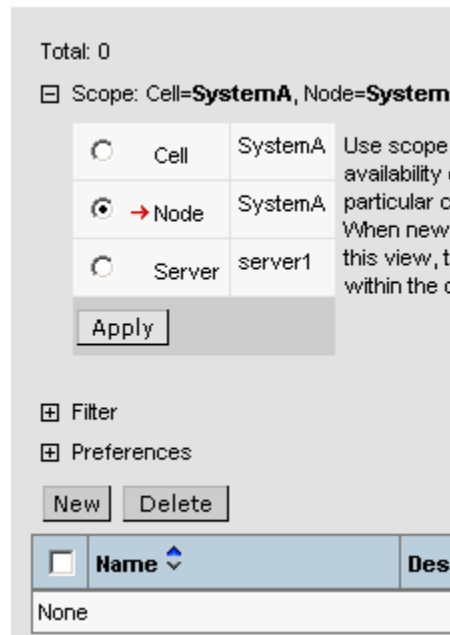
The User ID does not require a password, and does not need to be a User ID of a user in the local user registry. It is only used to track user-specific changes to configuration data. Security is NOT enabled

OK

- \_\_\_ c. Administration of WebSphere Application Server v5.0.1 is via a Web-based Administrative console. The console is basically composed of Navigation, Workspace, and System Tray panes. The Navigation pane on the left pane where different headings are listed for working with the different components and attributes of the server. Data sources and other resources are organized under the Resources heading. Expand Resources and click **JDBC Providers** to configure the data source which your MyBankCMP application will use to persist data.



- \_\_ d. The existing JDBC Providers will be displayed. You will be using DB2 as your JDBC Provider for the MyBank application. Verify that the Scope is set to Node, and then click **New** to create a DB2 Provider



- \_\_ e. In the main frame, select **DB2 Legacy CLI-based Type 2 JDBC Driver** from the drop down text box. Click **OK**.
- \_\_ f. The configuration properties for the JDBC Provider will be displayed. They are all correct. Notice that the classpath is a variable - **`\${DB2\_JDBC\_DRIVER\_PATH}/db2java.zip**. In the next step, we will update the WebSphere variable to point to the correct location on this node. Click **OK**.
- \_\_ g. Click **Environment**, then **Manage WebSphere Variables**.
- \_\_ h. Verify that the scope is set to the node level so that it matches the scope for the database driver (see Step \_\_d. Above).
- \_\_ i. Click on **DB2\_JDBC\_DRIVER\_PATH**

- \_\_ j. Enter the value **C:/SQLLIB/Java** . (Note: forward slashes.)

General Properties	
Name	* DB2_JDBC_DRIVER_PATH
Value	C:/SQLLIB/java
Description	The directory where contains the driver from DB2 JDBC2-compliant Provider.
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	

- \_\_ k. Click **OK**.
- \_\_ l. Return to JDBC Providers under Resources in the navigation pane.

Resources

[JDBC Providers](#)

- \_\_ m. With the JDBC provider created, you will now create a data source instance. Click **DB2 Legacy CLI-based Type 2 JDBC Driver**.

Name	Description
<a href="#">DB2 Legacy CLI-based Type 2 JDBC Driver</a>	DB2 JDBC2-compliant Provider

- \_\_ i. The JDBC Provider configuration for the DB2 Legacy CLI-based Type 2 JDBC Driver will be displayed again. Below the configuration, additional properties can be set. Click **Data Sources** (you may have to scroll down).

Additional Properties	
<a href="#">Data Sources</a>	Data Sour provider v
<a href="#">Data Sources (Version 4)</a>	This is the modules r

- \_\_ j. The empty list of Data Sources for the DB2 Legacy CLI-based Type 2 JDBC Driver will appear. Click **New** to create a Data Source.
- \_\_ k. The empty configuration properties for the Data Source will be displayed. Enter the following:

Name: **MyBankDS**  
 JNDI name: **jdbc/MyBank**  
 Select the checkbox next to **Use this DataSource in container managed persistence**  
 Description: **MyBank Datasource**

Configuration	
General Properties	
Scope	* cells: SystemA: nodes: SystemA
Name	* MyBankDS
JNDI Name	jdbc/MyBank
Container managed persistence	<input checked="" type="checkbox"/> Use this Data Source in container managed persistence (CMP)
Description	MyBank Datasource
Category	
Statement Cache Size	10 statements
Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2C
Component-managed Authentication Alias	
Container-managed Authentication Alias	
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	


**Note:** Ensure the box is checked for using the data source for container managed persistence. If this box is not checked, you will have problems at runtime (the correspondent connection factory is not going to be created).





- \_\_ l. Click **OK**.

- \_\_ m. The Data Source will be created, however there are a number of additional configuration properties which still need to be set. Click **MyBankDS**.

[JDBC Providers](#) > [DB2 Legacy CLI-based Type 2 JDBC Driver](#) >

## Data Sources

Data Source is used by the application to access the data from the database. A data source is created under a JDBC provider which provides the specific JDBC driver implementation class. 

Total: 1				
<input type="checkbox"/> Filter				
<input type="checkbox"/> Preferences				
<input type="button" value="New"/> <input type="button" value="Delete"/> <input type="button" value="Test Connection"/>				
<input type="checkbox"/>	Name 	JNDI Name 	Description 	Category 
<input type="checkbox"/>	<b>MyBankDS</b>	jdbc/MyBank	MyBank Datasource	

- \_\_ n. The Data Source configuration properties will be listed again. Below the configuration, additional properties can be set and you can work with related items. Under the related items header, click **J2C Authentication Data Entries** (you will have to scroll down).
- \_\_ o. Click **New**
- \_\_ p. Enter an alias of **MyBankAlias**, a user id of **wsdemo**, your password for the **wsdemo** account, and a description. Click **OK**.

Configuration	
<b>General Properties</b>	
Alias	* MyBankAlias
User ID	* wsdemo
Password	* *****
Description	MyBank Authentication Alias
<input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>	



\_\_ q. Take a short cut back to the datasource - click **MyBankDS**

[JDBC Providers](#) > [DB2 Legacy CLI-based Type 2 JDBC Driver](#) > [Data Sources](#) > **MyBankDS** >

### J2C Authentication Data Entries

\_\_ r. Scroll down and click **Custom Properties**, which is under the additional properties table.

\_\_ s. Click **databaseName**

\_\_ t. Change the Value to **BankData** and click **OK**.

General Properties	
Scope	* cells: SystemA; nodes: SystemA
Required	true
Name	databaseName
Value	BankData
Description	This is a required property. The database name.
Type	java.lang.String

Apply OK Reset Cancel

\_\_ u. Take the shortcut back to **MyBankDS**

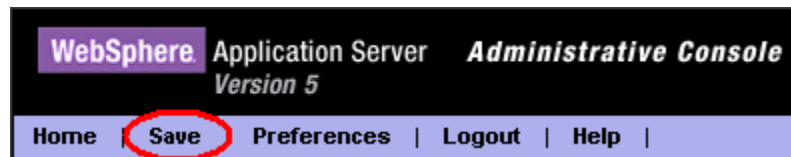
\_\_ v. From the Component Managed Authentication Alias drop-down, select the newly created **MyBankAlias** on your hostname and click **OK**

Datasource Helper Classname	com.ibm.websphere.rsadapter.DB2C
Component-managed Authentication Alias	SystemA/MyBankAlias
Container-managed Authentication Alias	

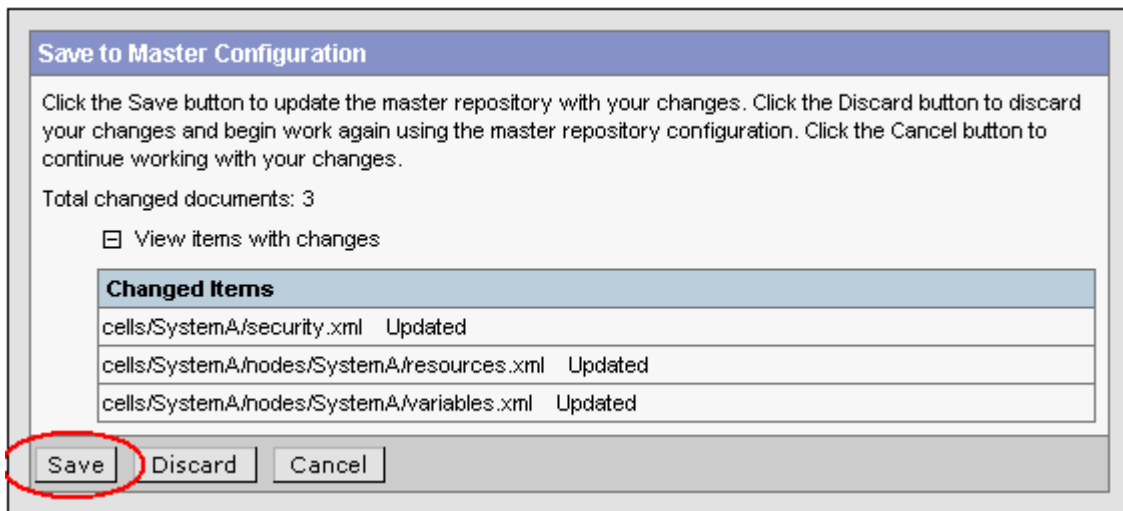
Apply OK Reset Cancel

\_\_ 3. Your Data Source is completely defined. With WebSphere Application Server v5.0, 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.

- \_\_ a. Click **Save** from the header.



- \_\_ b. The Save page will be displayed. Open View items with changes. The MyBank Authentication Alias you created is stored in security.xml, the datasource configuration in resources.xml, and the datasource classpath variable you edited will be in variables.xml. Click **Save**.

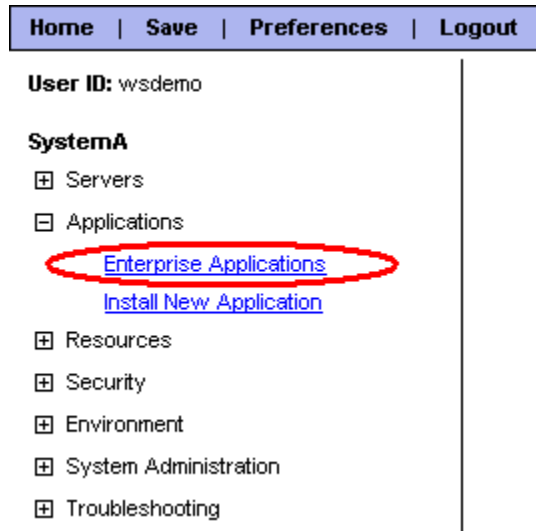


The WorkSpace Home will be displayed when the save operation completes. Before the database can be accessed by an application, the server will need to be restarted. Before restarting the application however, you will install the MyBankCMP application.

## **Part Two: Install Application into WebSphere and Create the Database**

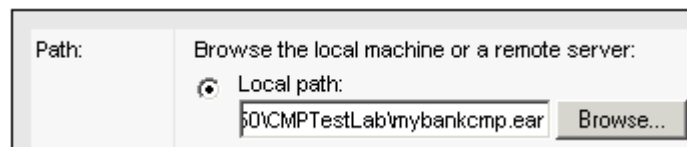
\_\_1. Install the MyBankCMP application.

\_\_ a. Expand **Applications** and click **Enterprise Applications**.



\_\_ b. The list of installed applications will be listed. Click **Install** to install the MyBankCMP application.

\_\_ c. You saved the application to your local system. Click **Browse...** for the local path and navigate to **C:\LabFiles50\CMPTTestLab\MyBankCMP.ear**



\_\_ d. Click **Next**.

- \_\_\_ e. Click **Next** again. The application installation will initialize. On the **Step 1** panel, make sure that the application name is MyBankCMP and that the "Deploy EJBs" box is checked.

→ **Step 1 : Provide options to perform the installation**

Specify the various options available to prepare and install your application.

AppDeployment Options	Enable
Pre-compile JSP	<input type="checkbox"/>
Directory to Install Application	<input type="text"/>
Distribute Application	<input checked="" type="checkbox"/>
Use Binary Configuration	<input type="checkbox"/>
Deploy EJBs	<input checked="" type="checkbox"/>
Application Name	<input type="text" value="MyBankCMP"/>
Create MBeans for Resources	<input checked="" type="checkbox"/>
Enable class reloading	<input type="checkbox"/>
Reload Interval	<input type="text"/>

- \_\_\_ f. Click **Next**. Step two has options to use when deploying enterprise beans. You should accept the defaults here. Click **Next**.
- \_\_\_ g. In Step three you can specify the JNDI names for the Account and Transfer enterprise beans. Verify that Account and Transfer have been set **ejb/MyBank/Account** and **ejb/MyBank/Transfer** respectively. Click **Next**.

JNDI Name
<input type="text" value="ejb/MyBank/Account"/>
<input type="text" value="ejb/MyBank/Transfer"/>

- \_\_\_ h. In Step four, you can specify a Data Source specifically for all the EJBs in your EJB jar file. Enter **eis/jdbc/MyBank\_CMP** for the JNDI name.
- \_\_\_ i. Click **Next**.
- \_\_\_ j. In Step 5, you can define a Data Source and a resource authorization for each entity EJB. We only have one Entity EJB, but will assign it a datasource to demonstrate the user interface. Expand **Apply Multiple Mappings**.

- \_\_ k. From the Specify existing Resource JNDI name drop down list, select **<SystemA>:eis/jdbc/MyBank\_CMP** (this is the connection factory that corresponds to the datasource we have created.) Also check the **EJB** checkbox. Click the associated (first) **Apply**. Verify that **eis/jdbc/MyBank\_CMP** appears under JNDI Name.

→ **Step 5: Map datasources for all 2.0 CMP beans**

Specify an optional data source for each 2.x CMP bean. Mapping a specific data source to a CMP bean will override the default data source for the module containing the Enterprise bean.

Apply Multiple Mappings

To apply multiple mappings, follow the steps below.

1. Select one or more checkboxes in the table
2. Complete mappings and click APPLY

Specify existing Resource JNDI name:

Resource authorization:

<input checked="" type="checkbox"/>	EJB	EJB Module	URI	JNDI Name	Resource Authorization
<input checked="" type="checkbox"/>	Account	MyBankCMPEJB	MyBankCMPEJB.jar,META-INF/ejb-jar.xml	<input type="text"/>	<input type="text" value="Container"/>

- \_\_ l. On the same panel, Select **Per Connection Factory** from the resource authorization drop down list and check the **EJB** checkbox again. Click the associated (second) **Apply**. Verify that the resource authorization for the Account EJB is Per Connection Factory.

- \_\_ m. Click **Next**

- \_\_\_ n. For Step six, you can specify EJB References which exist in your application. These references will need to match the JNDI names set for the referenced enterprise beans. Ensure the JNDI names match the graphic below and click **Next**.

→ **Step 6 : Map EJB references to beans**

Each EJB reference defined in your application must be mapped to an Enterprise bean.

Module	EJB	URI	Reference Binding	Class	JNDI Name
MyBankCMPEJB	Transfer	MyBankCMPEJB.jar,META-INF/ejb-jar.xml	bank/Account	com.ibm.mybank.ejb.AccountLocal	ejb/MyBank/Account
MyBankWeb		MyBankCMPWeb.war,WEB-INF/web.xml	ejb/Account	com.ibm.mybank.ejb.AccountLocal	ejb/MyBank/Account
MyBankWeb		MyBankCMPWeb.war,WEB-INF/web.xml	ejb/Transfer	com.ibm.mybank.ejb.TransferLocal	ejb/MyBank/Transfer

Previous Next Cancel

- \_\_\_ o. In Step seven, you can specify the virtual host which the Web module will be installed on. The default `_host` is the correct selection. Click **Next**.
- \_\_\_ p. For Step eight, you can specify which server you would like your application to be installed upon. Since you are working with only a single server, you do not have an option of a different server. Accept the default value and click **Next**.

**Note:** If you have multiple servers or want to be certain the server you will select the application upon, you can select the appropriate server from the dropdown list. For the base configuration of WebSphere which only has a single server, you do not need to specify the specific server.

- \_\_\_ q. In Step nine, security can be defined and set for specific methods within your EJB module. You have not set any security roles for the EJB module, click **Next** to move to the next step.
- \_\_\_ r. On step ten, click **Finish**. The installation will begin as the application is added to the server configuration and the enterprise beans are deployed. The progress will be displayed in the Workspace window of the console. When the installation has completed a success message will be listed. If a command prompt appears you can minimize the window. (It will close when EJBDeploy is finished.)

ADMA5013: Application MyBankCMP installed successfully.

Application MyBankCMP installed successfully.

- \_\_ s. When the application has finished installing, save the configuration changes by clicking **Save** on the heading. When the Save window appears, expand View items with changes. Notice the different items the configuration that will be updated with the installation of the application.

### Save to Master Configuration

Click the Save button to update the master repository with your changes. Click the Discard button to discard your changes and begin again using the master repository configuration. Click the Cancel button to continue working with your changes.

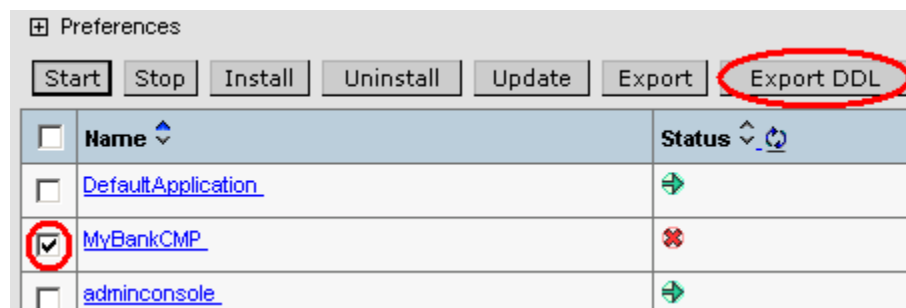
Total changed documents: 20

- View items with changes

#### Changed Items

cells/SystemA/nodes/SystemA/serverindex.xml	Updated
cells/SystemA/applications/MyBankCMP.ear/MyBankCMP.ear	Added
cells/SystemA/applications/MyBankCMP.ear/deployments/MyBankCMP/META-INF/ibm-application-bnd.xml	Added
cells/SystemA/applications/MyBankCMP.ear/deployments/MyBankCMP/MyBankCMP/Web.war/WEB-INF/web.xml	Added
cells/SystemA/applications/MyBankCMP.ear/deployments/MyBankCMP/MyBankCMPEJB.jar/META-INF/MANIFEST.MF	Added
cells/SystemA/applications/MyBankCMP.ear/deployments/MyBankCMP/MyBankCMP/Web.war/WEB-INF/ibm-web-bnd.xml	Add
cells/SystemA/applications/MyBankCMP.ear/deployments/MyBankCMP/MyBankCMPEJB.jar/META-INF/ibm-ejb-jar-ext.xml	Add

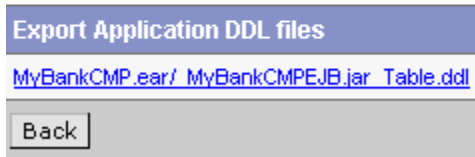
- \_\_ t. Scroll down and then click the **Save** button at the bottom.
- \_\_ 2. With the MyBankCMP application fully installed as well as the Data Source defined, you must complete the set up by creating the actual database which will be used to persist the account information.
- \_\_ a. From the navigator, select **Applications > Enterprise Applications**.
- \_\_ b. When WebSphere deployed the enterprise beans in your application, it generated a ddl file which contains the table create statement for actually creating the database table to use for persisting data from the account enterprise bean. The creation of this file only happens when the enterprise beans have been designed in a top-down fashion. This ddl file is stored with the application and can be retrieved through the administrative console. In the Application list, select the checkbox on the left of MyBank CMP and then click **Export DDL** from the buttons above the table.



- \_\_\_ c. The DDL file will be listed. Select **MyBankCMP.ear/\_MyBankCMPEJB.jar\_Table.ddl** to save the file to disk.

### Export Application DDL files

Click on the file name to download the DDL file



- \_\_\_ d. When prompted, choose to save the file to **c:\LabFiles50\CMPTestLab\table.ddl**.
- \_\_\_ e. Before actually creating the database, end your administration session. Log out of the console by selecting **Logout**.



- \_\_\_ f. We need to create the database and the table. In a DOS Command Prompt, navigate to **C:\LabFiles50\CMPTestLab\** and run the batch file **BankData.bat** using **wsdemo**, your wsdemo account password, and **table.ddl** as the three parameters.

Example: `C:\Labfiles50\CMPTestLab\>BankData.bat wsdemo password table.ddl`

**NOTE:** The extension is .DDL, not .DLL. It stands for Database Definition Language.

- \_\_\_ g. Even though the BankData script reports success, view the file **C:\LabFiles50\CMPTestLab\BankDataLoadDB.log**. If there are errors, correct them and re-run the previous step. (If wsdemo.account and wsdemo.customer are undefined, this is normal - those tables do not exist the first time this script runs.)
- \_\_\_ h. Once the database has been created, you must stop and restart the server for the Data Source to be properly registered in JNDI and bound to the application server. Change to the **\WebSphere\Appserver\bin** directory and stop and restart the server.



```
cd \websphere\appserver\bin
```

```
stopserver server1
```

```
ADMU0116I: Tool information is being logged in file  
           C:\WebSphere\AppServer\logs\server1\stopServer.log
```

```
ADMU3100I: Reading configuration for server: server1
```

```
ADMU3201I: Server stop request issued. Waiting for stop status.
```

```
ADMU4000I: Server server1 stop completed.
```

```
startserver server1
```

```
ADMU0116I: Tool information is being logged in file  
           C:\WebSphere\AppServer\logs\server1\startServer.log
```

```
ADMU3100I: Reading configuration for server: server1
```

```
ADMU3200I: Server launched. Waiting for initialization status.
```

```
ADMU3000I: Server server1 open for e-business; process id is 2880
```

### \_\_3. Test the JDBC Connection to the Datasource

- \_\_ a. If a window with an admin console login prompt is not already open, go to **http://localhost:9090/admin**. Log in as **wsdemo**.
- \_\_ b. Under **Resources**, click **JDBC Providers**
- \_\_ c. Click on **DB2 Legacy CLI-based Type 2 JDBC Driver**
- \_\_ d. Scroll down and click on **Data Sources**
- \_\_ e. Check the box by **MyBankDS** and click **Test Connection**. The test should be successful.
- \_\_ f. Click the **Logout** link on the top of the screen.

### **Part Three: Testing the Application**

- \_\_1. Use a web browser to access the MyBankCMP application at <http://localhost:9080/MyBankCMPWeb/index.html>.
- \_\_2. Click **Create an Account** link.

- \_\_3. For the Account Number enter **1** and for the Starting Balance enter **1000** and click **Create**.

## Create a new Account

Account Number:	
Type:	<input checked="" type="radio"/> savings <input type="radio"/> checking
Starting Balance:	
<input type="button" value="Create"/>	
Created account: 1, Type: savings, Balance: 1000	

- \_\_4. Create a second account with a number of **2** and a balance of **4000**.
- \_\_5. Click the **Transfer Funds** link.
- \_\_6. Enter **200** for the Amount and **1** for the From Account and **2** for the To Account. Click **Transfer**.

## Transfer Funds between Accounts

Amount	From Account	To Account
	1	2
	Balance	Balance
<input type="button" value="Transfer"/>	800.00	4200.00
<input type="button" value="Clear"/>	<input type="button" value="Get Balance"/>	<input type="button" value="Get Balance"/>

Transfer successful

### **Part Four: Lab Clean Up - IMPORTANT**

- \_\_1. After completing the testing of this lab, it is necessary to remove the application from the server. You can uninstall the application from the administration console (remember to perform a Save after uninstalling), however, you will use the wsadmin command line

interface instead. From a command prompt change to the bin directory underneath the WebSphere installation root directory.

```
C:\>cd WebSphere\Appserver\bin
```

- \_\_2. Execute the command `wsadmin -c "$AdminApp uninstall MyBankCMP"`. This will remove the MyBankCMP application from your server using `wsadmin`, the new command-line interface for WebSphere.

```
C:\WebSphere\Appserver\bin>wsadmin -c "$AdminApp uninstall MyBankCMP"
WASX7209I: Connected to process "server1" on node ServerA using SOAP
connector;
The type of process is: UnManagedProcess
ADMA5017I: Uninstallation of MyBankCMP started.
ADMA5104I: Server index entry for ServerA was updated successfully.
ADMA5102I: Deletion of config data for MyBankCMP from config repository
complete
d successfully.
ADMA5011I: Cleanup of temp dir for app MyBankCMP done.
ADMA5106I: Application MyBankCMP uninstalled successfully.
```

- \_\_3. Stop your server.

```
C:\WebSphere\Appserver\bin>stopserver server1
```

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

In this exercise you, configured your WebSphere server with a JDBC Provider and a data source using the Administrative console. You then installed a J2EE 1.3 application using the Administrative console. After the application was installed, you configured the database and then started the server to test the application using a Web browser.

This page is intentionally left blank.