

## IBM RATIONAL APPLICATION DEVELOPER 6.0 – LAB EXERCISE

# Testing MDB Application in IBM Rational Application Developer

What this exercise is about .....	1
Lab Requirements.....	1
What you should be able to do .....	1
Introduction .....	2
Part 1: Server Setup.....	5
Part 2: Application and Cloudscape Setup .....	6
Part 3: Preparing the JMS Resources .....	8
Part 4: Initializing database components .....	14
Part 5: Test Application.....	18
Part 6: Restore your server configuration .....	22
What you did in this exercise .....	23

## What this exercise is about

This exercise will test an MDB(Message Driven Bean). The application with the MDB will be deployed to the server. The JMS(Java Message Service) will be used to send a message from the client running in IBM Rational Application Developer to the server, where the MDB will be invoked to process the message request.

## Lab Requirements

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

- Windows 2000 Professional Service Pack 4 is required for this lab exercise.
- IBM Rational Application Developer v6.0, with the WebSphere Application Server v6.0 Integrated Test Environment is also required.
- The lab source files (LabFiles60.zip) must be extracted to the root directory (i.e., C:\).
- Experience with previous versions of Rational Application Developer and the J2EE programming model are also required.

## What you should be able to do

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

- Prepare a MDB application for testing on the Integrated Test Environment Application Server.
- Set up Cloudscape as a data source for your application before testing the application.
- Set up JMS.
- Test your application on the Integrated Test Environment Application Server.

---

## Introduction

The J2EE 1.4 application which you will be testing is a simple banking application named WebSphereBank. It is composed of MDB, EJB, Web, and Application Client modules. In the EJB module there is a session bean and an entity bean. Before being able to fully test the application, you will need to generate mappings for the entity bean to a data source as well as generate deploy and RMIC code used by the EJB container of the J2EE server to handle the calls to the entity bean and persisting of the data. Cloudscape 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. You will also need to perform some setup of the JMS resources.

In this lab, you will be deploying the WebSphereBank application on the Integrated Test Environment application server. Finally, you will test the application by initiating a message from the client to the Integrated Test Environment application server.

---

### Exercise Instructions

**\*\* NOTE \*\* Solution instructions are normally provided at the end. The solution is not provided in this case because you need to do the exercises in order to understand Java tools and there is no final solution to import. To go through the lab, start at Part One assuming you have met the requirements in the section “User Requirements” stated above.**

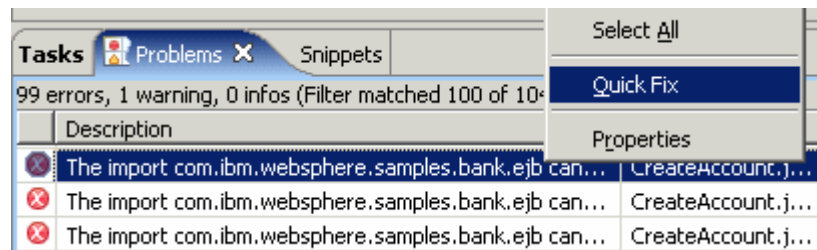
Reference Variable	Windows Location	AIX/UNIX Location
<LAB_FILES>	C:\Labfiles60	/tmp/Labfiles60Beta2
<LAB_NAME>	IRAD_TestingMDBLocal	TestingMDBLocal
<RAD_HOME>	C:\Program Files\IBM\RSDP\6.0	

## Part 1: Server Setup

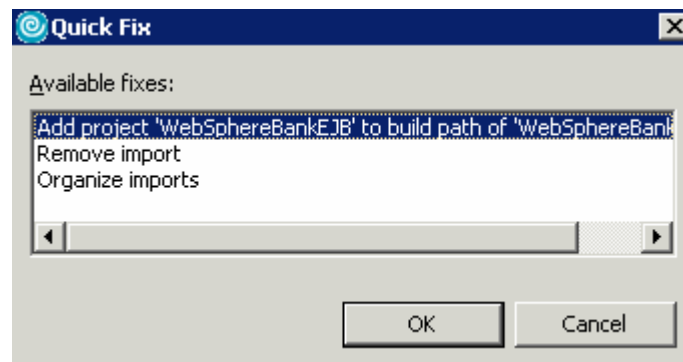
- \_\_\_ 1. Backup your server configuration. This will preserve your current server configuration. You will restore your server configuration at the end of the lab exercise.
  - \_\_\_ a. Open a Windows **Command Prompt** and navigate to the following directory:  
`<RAD_HOME>\runtimes\base_v6\bin`
  - \_\_\_ b. Backup the server configuration by issuing the following command:  
`backupConfig "c:\Program Files\IBM\Backup.zip"`
- \_\_\_ 2. Start Rational Application Developer.
  - \_\_\_ a. **Start > Programs > IBM Rational > IBM Rational Application Developer V6.0 > Rational Application Developer.**
  - \_\_\_ b. For workspace, specify `<LAB_FILES>\IRAD_TestMDBLocal\workspace`.
- \_\_\_ 3. Open the J2EE Perspective if it is not already open
  - \_\_\_ a. Close the welcome screen.
  - \_\_\_ b. From the menu select **Window > Open Perspective > Other.**
  - \_\_\_ c. Select J2EE and click **OK.**

## Part 2: Application and Cloudscape Setup

- \_\_\_ 1. Back in IBM Rational Application Developer, import the WebSphereBank application for testing.
  - \_\_\_ a. Select **File > Import...**
  - \_\_\_ b. Select **EAR file** and select **Next**.
  - \_\_\_ c. Select **Browse...** and navigate to <LAB\_FILES>IRAD\_TestMDBLocal\WebSphereBank.ear and select **Open**.
  - \_\_\_ d. For the Project name, enter **WebSphereBank**.
  - \_\_\_ e. Click **Finish**.
- \_\_\_ 4. Add Java Build Path to clean up errors. If you select the Problems tab, you will notice a list of errors and warnings.
  - \_\_\_ a. Right click on an error and select **Quick Fix**.



- \_\_\_ 5. A list of available fixes will appear. Select "Add project 'WebSphereBankEJB' to build path of 'WebSphereBankWeb'" and select **OK**. The errors should disappear.



- \_\_\_ 6. Start WebSphere Application Server.
  - \_\_\_ a. Select the server to be started, on the Servers tab, then click on the **Start the server** icon.



\_\_\_ b. When the server start has completed, you should see the following information:

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


---

**NOTE:** The server may also be started from the command line by navigating to the **C:\Program Files\IBM\RADBeta6.0\runtimes\base\_v6\bin** directory and issuing the **startserver server1** command.

---

## Part 3: Preparing the JMS Resources

- \_\_\_ 1. Log on to the Administrative Console
  - \_\_\_ a. In a browser, enter the URL: <http://localhost:9060/ibm/console>
  - \_\_\_ b. Any string can be entered as the user ID (e.g.: Enter **wsdemo** and click **Log In**)

- \_\_\_ 7. Create a SIBus. To see an animation , click here: 
  - \_\_\_ c. In the Administrative Console, expand **Service Integration**. Click **Buses**
  - \_\_\_ d. Click **New** to create a new bus
  - \_\_\_ e. Enter the name as **mybus**

\* Name

UUID

Description

Secure

Inter-engine authentication alias

Mediations authentication alias

Protocol

Discard messages after queue deletion

Configuration Reload Enabled

- \_\_\_ f. Click **OK**.
- \_\_\_ 8. Save changes
  - \_\_\_ g. In the Messages area, click **Save**
  - \_\_\_ h. Click **Save** again to save changes to master configuration
- \_\_\_ 9. Make server1 a member of mybus
  - \_\_\_ i. Under **Service Integration**, click **Buses**
  - \_\_\_ j. Click **mybus**

- \_\_\_ k. Under Additional Properties, select **Bus Members**
- \_\_\_ l. Click **Add** to add a new bus member
- \_\_\_ m. Make sure that the **Server** radio button is selected
- \_\_\_ n. From the server selection box, select **<mynodename>:server1**

Add an application server or server cluster as a new member of the bus.

- \_\_\_ o. Click **Next**
- \_\_\_ p. Click **Finish**

10. Let's now create the SIB destination. In this exercise, we are using point-to-point messaging, ie, our MDB needs to listen to a queue. So, let's create a SIB queue

- \_\_\_ a. In the Administrative Console under **Service Integration**, click **Buses**.
- \_\_\_ b. Click on **mybus**.
- \_\_\_ c. In the **Additional Properties**, click **Destinations**.
- \_\_\_ d. Click **New** to create a new destination.
- \_\_\_ e. Select **Queue** as the Destination Type and click **Next**.
- \_\_\_ f. In the Set Queue Attributes panel, enter the Identifier as **BankJSQueue**.

Create a new queue for point-to-point messaging

\_\_\_ g. Click **Next**.

\_\_\_ h. In the Assign Bus member panel, server1 is selected by default. Click **Next**.

\_\_\_ i. Click **Finish** on the Confirmation Panel.

\_\_\_ 11. Save changes

\_\_\_ a. In the Messages area, click **Save**

\_\_\_ b. Click **Save** again to save changes to master configuration.

\_\_\_ 12. Let's now create the JMS resources that will point at the SIB Destination. Create a JMS Connection factory.

\_\_\_ a. Expand **Resources > JMS Providers**. Click **Default messaging**.

\_\_\_ b. Under **Connection Factories**, click **JMS Connection Factory**.

\_\_\_ c. Click **New**.



\_\_\_ d. Enter the following:

Name: **BankJMSSConnFactory**  
JNDI Name: **jms/BankJMSSConnFactory**  
Bus Name: **mybus**

Leave all the other values to the default.

### General Properties

\* Scope

\* Name

\* JNDI name

Description

Category

Bus name

Client identifier

\* Non-persistent message reliability

\* Enable message streaming

Temporary queue name prefix

Temporary topic name prefix

Durable subscription home

\* Share durable subscriptions

Remote target group

\_\_\_ e. Select **Apply**, then **OK**.

\_\_\_ 13. Create a JMS Queue.

\_\_\_ a. Under **Resources > JMS Providers**. Click **Default messaging**.

\_\_\_ b. Under **Destinations**, click **JMS Queue**. Click **New**.

- \_\_\_ c. Enter the following:  
Name: **BankJMSQueue**  
JNDI Name: **jms/BankJMSQueue**  
Queue Name: **BankJSQueue**  
Bus Name: **mybus**

Leave all the other values to default.

### General Properties

\* Scope

\* Name

\* JNDI name

Description

\* Queue name

Delivery mode

Time to keep messages

Priority

Enable message streaming

Bus name

- \_\_\_ d. Select **Apply**, then **OK**.

\_\_\_ 14. Create an activation spec for the Message Driven Bean.

- \_\_\_ a. Under **Resources > JMS Providers**. Click **Default Messaging Provider**.
- \_\_\_ b. Under **Activation Specs**, click **JMS activation specification**. Click **New**.
- \_\_\_ c. Enter the following:  
Name: **BankActivationSpec**  
JNDI Name: **eis/BankActivationSpec**  
Destination JNDI Name: **jms/BankJMSQueue**  
Bus Name: **mybus**

Leave all the other values to default.


- \_\_\_ d. Select **Apply**, then **OK**.

\_\_\_ 15. Save changes.

\_\_\_ a. In the Messages area, click **Save**.

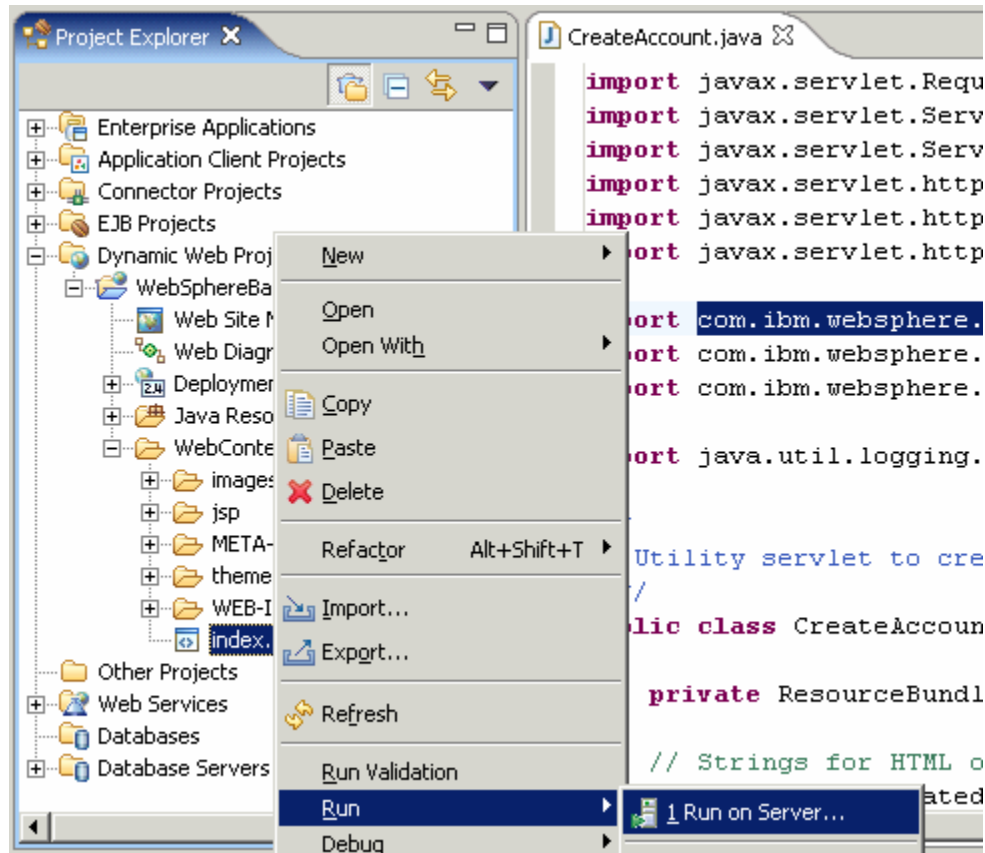
\_\_\_ b. Click **Save** again to save changes to master configuration.

\_\_\_\_ 16. Restart the server.

\_\_\_ a. From server window in Rational Application Developer, click on the server and then click on the recycle server icon .

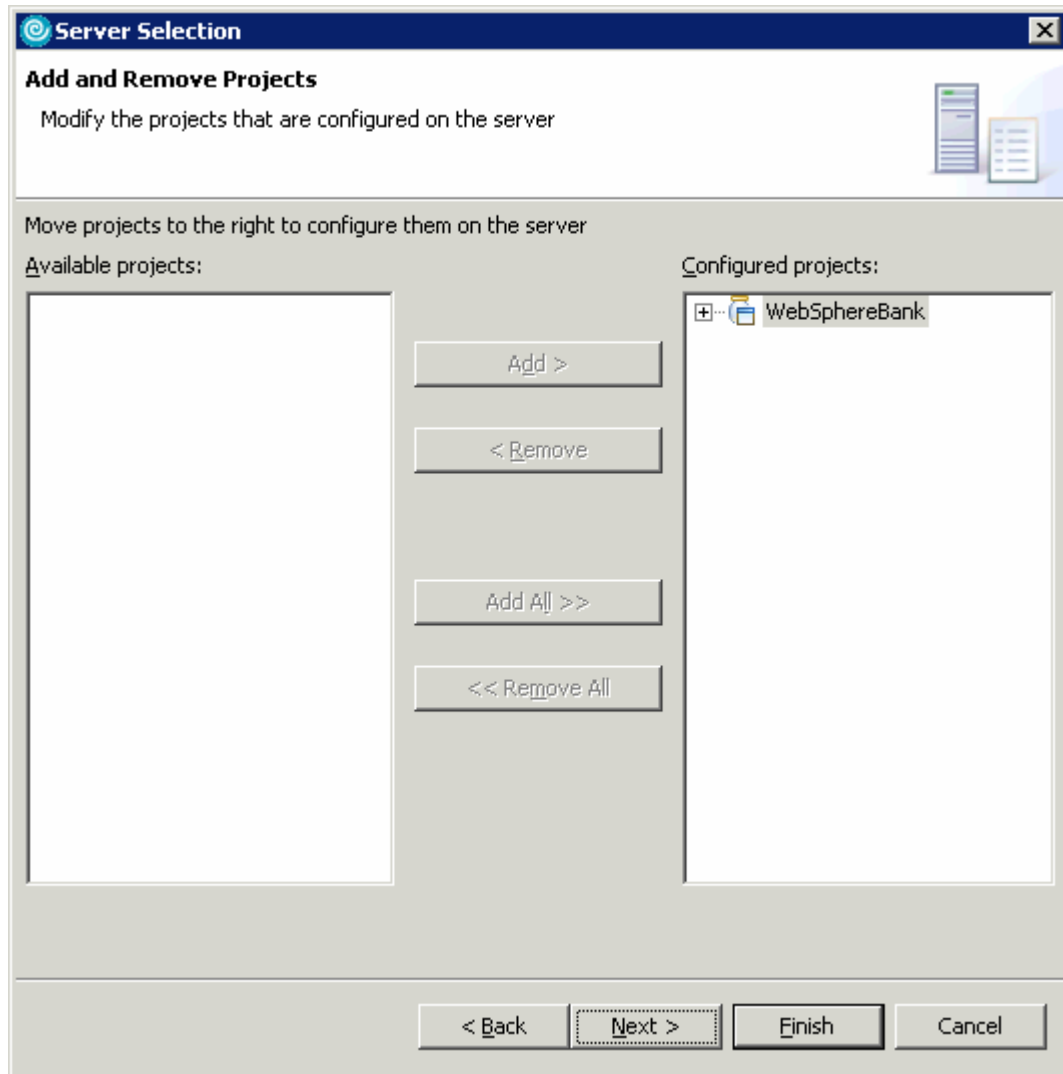
## Part 4: Initializing database components

- \_\_\_ 1. Start the server with the WebSphereBank project while initializing the database and datasource.
  - \_\_\_ a. In the Project Explorer view, navigate to **Dynamic Web Projects > WebSphereBankWeb > WebContent** and right click on index.html.

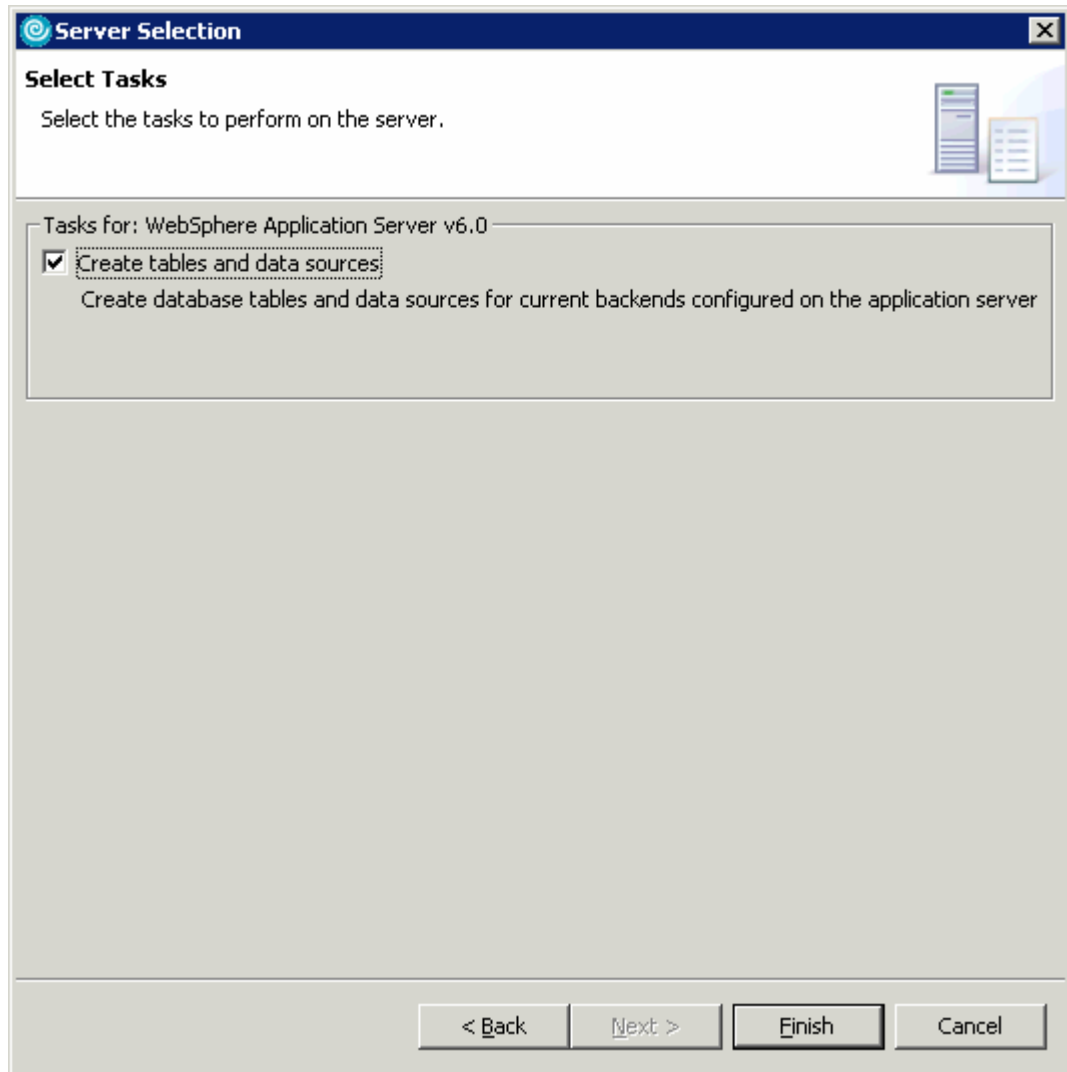


- \_\_\_ b. Select the **Run > Run on Server** option to open the Server Selection window.
- \_\_\_ c. On the Define a New Server page of the Server Selection window make sure the **Choose an existing server** option is selected as well as the server WebSphere Application Server v6.0. Then click on the **Next** button.

- \_\_\_ d. On the **Add and Remove Projects** page, make sure that WebSphereBank is added to the Configured projects list on the right side of the window. Then click on the **Next** button.

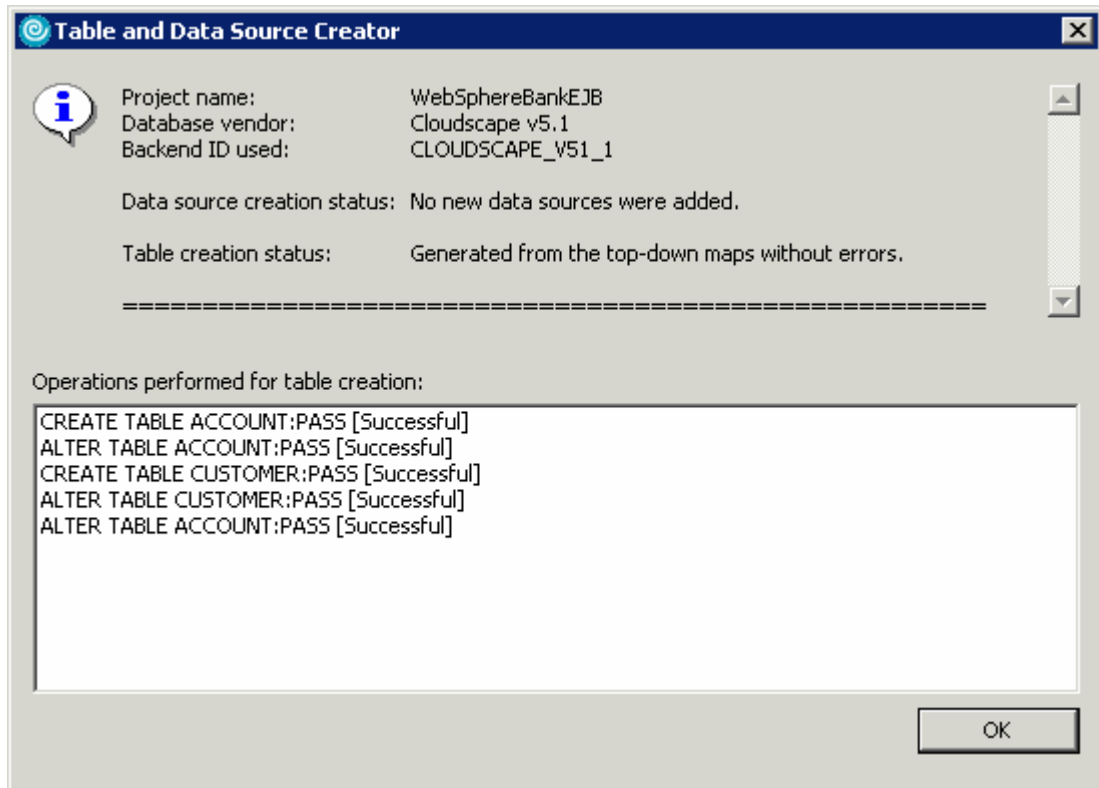


- \_\_\_ e. On the **Select Tasks** page, click on the Create tables and data sources checkbox. Then click the **Finish** button.



- \_\_\_ f. Click **Finish**.

\_\_ g. You should see a window like the one below if the database has been setup successfully.



---

**NOTE:** After the code has been generated, browse to EJBProjects > WebSphereBankEJB > ejbModule. Notice the classes and the packages which were created. In the `com.ibm.websphere.samples.bank.ejb.websphere_deploy` package are common classes used to support persisting entity beans running in WebSphere regardless of the data source. The `com.ibm.websphere.samples.bank.ejb.websphere_deploy.CLOUDSCAPE_V5_1` classes are specific for persisting entity beans to a Cloudscape database.

---

\_\_ h. Click **OK**.

## Part 5: Test Application

- \_\_\_ 1. After the application has made it to the server, open your browser to the WebSphereBank homepage. Open a Web Browser and navigate to the following URL:

**http://localhost:9080/WebSphereBankWeb**

It may take some time, but eventually the homepage of the WebSphereBank Application will come up. The WebSphereBank Application is composed of several processes (Create Customer, Create Account, Transfer funds, etc). First you will test the application by creating a customer, then creating two accounts before transferring funds between the two accounts.

- \_\_\_ 2. Select the **Create Customer** link.
- \_\_\_ 3. Enter Customer Number, Name and Tax ID. Select **Create**.

### Create Customer

---

Messages

---

---

Customer Number:	<input type="text" value="10"/>
First Name:	<input type="text" value="John"/>
Last Name:	<input type="text" value="Doe"/>
TAX ID:	<input type="text" value="012-34-5678"/>



- \_\_\_\_ 4. You will see details for customer created. Select **Create Account**.

## Customer Details

New Customer has been successfully created

Customer Number:	10
First Name:	JOHN
Last Name:	DOE
TAX ID:	012-34-5678

Create Account

- \_\_\_\_ 5. Enter **101** for the Account Number, checking for account type and **600** for the starting balance. Select **Create**.

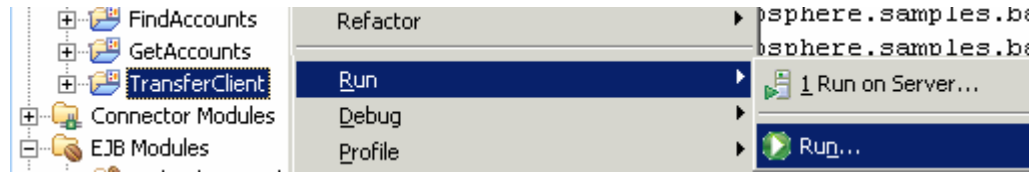
## Create a new Account

Messages

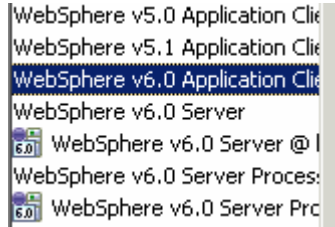
Customer Number:	<input type="text" value="10"/>
Account Number:	<input type="text" value="101"/>
Account Type:	<input type="radio"/> Savings <input checked="" type="radio"/> Checking
Starting Balance:	\$ <input type="text" value="600"/>

- \_\_\_\_ 6. Create a savings account with a number of **102** and a balance of **400**.
- \_\_\_\_ 7. Run Application Client Code.
- \_\_\_ a. In IBM Rational Application Developer, project explorer view, expand **Application Client Projects**.

\_\_ b. Right click on TransferClient and select **Run > Run** from the context menu.



\_\_ c. In the configurations list select WebSphere Application Server v6 Application client and select **New**.



\_\_ d. Make sure the **Application client module** is **TransferClient**.

\_\_ a. In the **Classpath** tab, add **ibmorb.jar** to the list of External Jars

1) Click on Bootstrap entries and select **Add External Jars**.

2) Navigate to <RAD\_HOME>\runtimes\base\_v6\java\jre\lib\ibmorb.jar. Select **Open**.

\_\_ b. On the Arguments tab you should enter the following in the Program Arguments text area:

-CCverbose=true 101 102 50

\_\_ c. Click **Run**. This will transfer \$50 from account 101 to account 102.

\_\_\_ 12. Go to the WebSphereBank web interface and verify the transfer.

\_\_ a. Select **Get Balance** function.

\_\_ b. Enter **101** for account and select **Balance**.

\_\_ c. Message will indicate account balance of 550.

## Get Balance

---

Messages

The Balance for the Account Number 101 is \$550.00

Account Number:

\_\_\_ d. Repeat for account **102**.

\_\_\_ 13. The application works properly. Close the Web **Browser**.

\_\_\_ 14. Select the **Servers** view from the Server Perspective and stop the server. Select **server1** and right-click selecting **Stop**.

## Part 6: Restore your server configuration

\_\_\_ 1. Restore your server configuration. This will return your server configuration to its original state.

\_\_\_ a. Open a Windows **Command Prompt** and navigate to the following directory:

```
<RAD_HOME>\runtimes\base_v6\bin
```

\_\_\_ b. Restore the server configuration by issuing the following command:

```
restoreConfig "c:\Program Files\IBM\Backup.zip"
```

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

In this exercise, you tested an application containing a Message Driven Bean.

This page is left intentionally blank.