How to Run 4-Tier KWIC Project on J2EE Platform

1. Preparation

1. Start Cloudscape database server.
   ```
   C:\> cloudscape –start
   ```
2. Build the database "webaddressaccount".
   ```
   C:\> cloudscape –isql
   ```
   At the database server's input prompt ij> type in SQL command. These command are saved in 4tier\example\src\sql\WebAddressAccount.sql.

   ```
    create table webaddressaccount
   (
     urlName varchar(40)  constraint pk_webaddress_account primary key,
     urlDescript varchar(240)
   );
   exit;
   ```

3. Start J2EE server.
   ```
   C:\> j2ee -verbose
   ```
4. Start deploytool.
   ```
   C:\> deploytool
   ```

2.Creating the J2EE Application

The sample application contains two J2EE components: an enterprise bean and a Web component. Before building these components, you will create a new J2EE application called ProjectApp and will store it in an EAR file named ProjectApp.ear.
Steps:
1. In deploytool, select File->New->Application.   Click Browse.
2. In the file chooser, navigate to 4tier/example/build.
3. In the File Name field, enter ProjectApp.ear.
4. Click New Application. Click OK.

3. Creating the Enterprise Bean

An enterprise bean is a server-side component that contains the business logic of an application. At runtime, the application clients execute the business logic by invoking the enterprise bean's methods. The enterprise bean in our example is a entity bean called WebAddressAccountEJB. The source code for WebAddressAccountEJB is in the 4tier/examples/src/ejb directory.

Codes of the Enterprise Bean:
The enterprise bean in this example requires the following code:
Remote interface: WebAddressAccount.java
Home interface: WebAddressAccountHome.java
Enterprise bean class: WebAddressAccountBean.java

Compiling the Source Files
Now you are ready to compile these classes, go to the 4tier/examples/src/ejb/ directory and type: javac *.java

Packaging the Enterprise Bean
To package an enteprise bean, you run the New Enterprise Bean wizard of the deploytool utility. During this process, the wizard performs the following tasks.

- ✓ Creates the bean's deployment descriptor
- ✓ Packages the deployment descriptor and the bean's classes in an EJB JAR file
- ✓ Inserts the EJB JAR file into the application's ProjectApp.ear file
- ✓ During the packaging process, you can view the deployment descriptor by selecting ToolsDescriptor Viewer.

To start the New Enterprise Bean wizard, select File->New->Enterprise Bean. The wizard displays the following dialog boxes.

1. Introduction dialog box
a) Read the explanatory text for an overview of the wizard's features.
b) Click Next.

2. EJB JAR dialog box
a) Select the Create New JAR File In Application button.
b) In the combo box, select ProjectApp.
c) In the JAR Display Name field, enter WebAddressAccountJAR
d) Click Edit.
e) In the tree under Available Files, locate the 4tier/example/src/ejb directory. (If the directory is many levels down in the tree, you can simplify the tree view by entering all or part of the directory's path name in the Starting Directory field.)
f) Select the following classes from the Available Files tree and click Add: WebAddressAccount.class, WebAddressAccountHome.class, and WebAddressAccountBean.class. (You may also drag and drop these class files to the Contents text area.)
g) Click OK.
h) Click Next.

3. General dialog box
a) Under Bean Type, select the "Entity" radio button.
b) In the Enterprise Bean Class combo box, select WebAddressAccountBean.
c) In the Enterprise Bean Name field, enter WebAddressAccountEJB.
d) In the Remote Home Interface combo box, select WebAddressAccountHome.
e) In the Remote Interface combo box, select WebAddressAccount.
f) Click Next.

4. Transaction Management dialog box

In the Entity Settings dialog box, select the radio button for Bean-Managed Persistence.

Because you may skip the remaining dialog boxes, click Finish.

Specifying a Resource Reference

1. In deploytool, select WebAddressAccountEJB from the tree.
2. Select the Resource Refs tab.
3. Click Add.
4. In the Coded Name field, enter jdbc/WebAddressAccountDB.
   The WebAddressAccountBean code refers to the database as follows:

```
private String dbName = "java:comp/env/jdbc/WebAddressAccountDB";
```

The java:comp/env prefix is the name of the JNDI context for the component. The jdbc/WebAddressAccountDB string is the JNDI name for the resource reference. The JNDI names for JDBC DataSource objects are stored in the java:comp/env/jdbc subcontext.

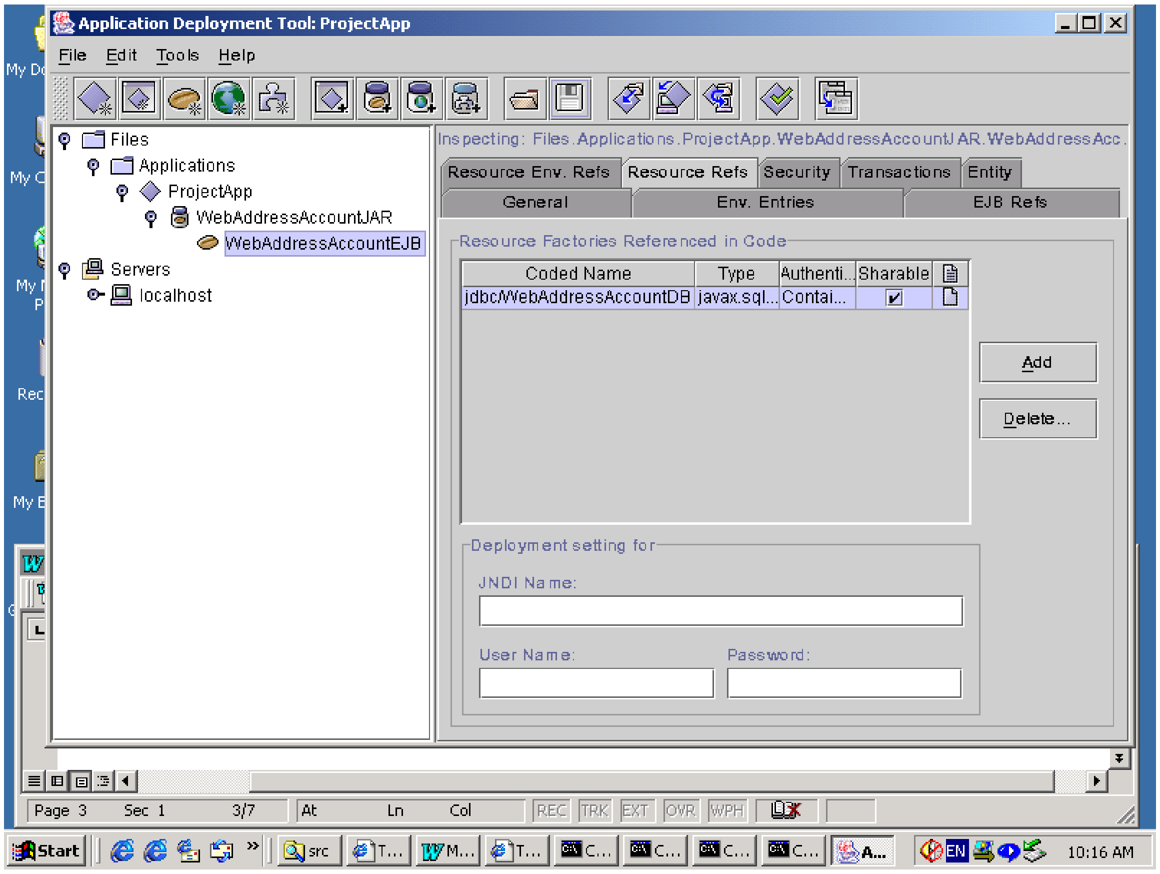
5. In the Type combo box, select javax.sql.DataSource. A DataSource object is a factory for database connections.
6. In the Authentication combo box, select Container.
7. If you want other enterprise beans to share the connections acquired from the DataSource, select the Sharable checkbox.

If the preceding steps are followed, the Resource Refs tab will appear as shown in

Figure-1.

Mapping a Resource Reference to a JNDI Name

1. Select the ProjectApp application from the tree.
2. Select the JNDI Names tab.
3. In the References table, select the row containing the resource reference. The reference name is jdbc/WebAddressAccountDB, the name you entered in the Coded Name field of the Resource Refs tab.
4. In the row you just selected, enter the JNDI name. For the WebAddressAccountEJB example, you would enter jdbc/Cloudscape in the JNDI Name field.



Inspecting: Files.Applications.ProjectApp.WebAddressAccountJAR.WebAddressAcc.

Resource Factories Referenced in Code

Coded Name	Type	Authenti...	Sharable
jdbc/WebAddressAccountDB	javax.sql...	Contai...	✓

Deployment setting for

JNDI Name:

User Name:

Password:

Figure-1 Resource Refs Tabbed Pane of WebAddressAccountEJB

4. Creating the Web Client

The Web client is contained in the JSP page, example/src/web directory. A JSP page is a text-based document that contains static template data, which can be expressed in any text-based format such as HTML, WML, and XML; and JSP elements, which construct dynamic content.

Codes of the Web Client

The statements (in bold in the following code) for locating the home interface, creating an enterprise bean instance, and invoking a business method are nearly identical to those of the J2EE application client.

The classes needed by the client are declared with a JSP page directive (enclosed within the <%@ %> characters). Because locating the home interface and creating the enterprise bean are performed only once, this code appears in a JSP declaration (enclosed within the <%! %> characters) that contains the initialization method, jspInit, of the JSP page. The declaration is followed by standard HTML markup for creating a form with an input field. A scriptlet (enclosed within the <% %> characters) retrieves a parameter from the request and converts it to a double. Finally, JSP expressions (enclosed within <%= %> characters) invoke the enterprise bean's business methods and insert the result into the stream of data returned to the client.

JSP Pages:
- ✓ Alphabetizer.jsp
- ✓ CshiftLines.jsp
- ✓ FindHandler.jsp
- ✓ SaveHandler.jsp

Other HTML Pages:
- ✓ MenuFrame.html
- ✓ Save.html
- ✓ Find.html
- ✓ Welcome.html
- ✓ Index.html

Java Bean Classes
- ✓ Alphabetizer.java
- ✓ CshiftLines.java
- ✓ DetailURL.java
- ✓ KeyWordSearch.java
- ✓ Line.java
- ✓ LinkedQueue.java
- ✓ URLEventObject.java
- ✓ URLListener.java

Compiling the Web Client
The J2EE server automatically compiles Web clients that are JSP pages. For the Java Bean classes, you would have to compile it.

Now you are ready to compile these classes, go to the 4tier/examples/src/classes/ directory and type: javac *.java

Packaging the Web Client
To package a Web client, you run the New Web Component wizard of the deploytool utility. During this process the wizard performs the following tasks.

- ✓ Creates the Web application deployment descriptor
- ✓ Adds the component files to a WAR file
- ✓ Adds the WAR file to the application's ConverterApp.ear file

During the packaging process, you can view the deployment descriptor by selecting Tools->Descriptor Viewer.

To start the New Web Component wizard, select File->New->Web Component. The wizard displays the following dialog boxes.

1. Introduction dialog box
a) Read the explanatory text for an overview of the wizard's features.
b) Click Next.
2. WAR File dialog box
a) Select Create New WAR File In Application.
b) In the combo box, select ProjectApp.
c) In the WAR Display Name field, enter ProjectWAR.
d) Click Edit.
e) In the tree under Available Files, locate the 4tier/example/src/web directory.
f) Select index.html and click Add. Continue to add all jsp pages and html pages.
g) In the tree under Available Files, locate the 4tier/examples/src/classes directory.
h) Select all class files and click Add.
i) Click OK.
j) Click Next.
3. Choose Component Type dialog box
a) Select the JSP radio button.
b) Click Next.
4. Component General Properties dialog box
a) In the JSP Filename combo box, select SaveHandler.jsp.
b) Click Finish.

Specifying the Web Client's Enterprise Bean Reference
When it invokes the lookup method, the Web client refers to the home of an enterprise bean:

```
Object objRef = ic.lookup("java:comp/env/ejb/TheWebAddressAccount ");
```

You specify this reference as follows:

1. In the tree, select ProjectWAR.
2. Select the EJB Refs tab.
3. Click Add.
4. In the Coded Name column, enter ejb/TheWebAddressAccount.
5. In the Type column, select Entity.

6. In the Interfaces column, select Remote.
7. In the Home Interface column, enter WebAddressAccountHome.
8. In the Local/Remote Interface column, enter WebAddressAccount.

Specifying the JNDI Names
When the Web client access the same enterprise bean, their code refers to the bean's home by ejb/TheWebAddressAccount. This references are in the parameters of the lookup calls. In order for the lookup method to retrieve the home object, you must map the references in the code to the enterprise bean's JNDI name. Although this mapping adds a level of indirection, it decouples the clients from the beans, making it easier to assemble applications from J2EE components.

To map the enterprise bean references in the clients to the JNDI name of the bean, follow these steps.

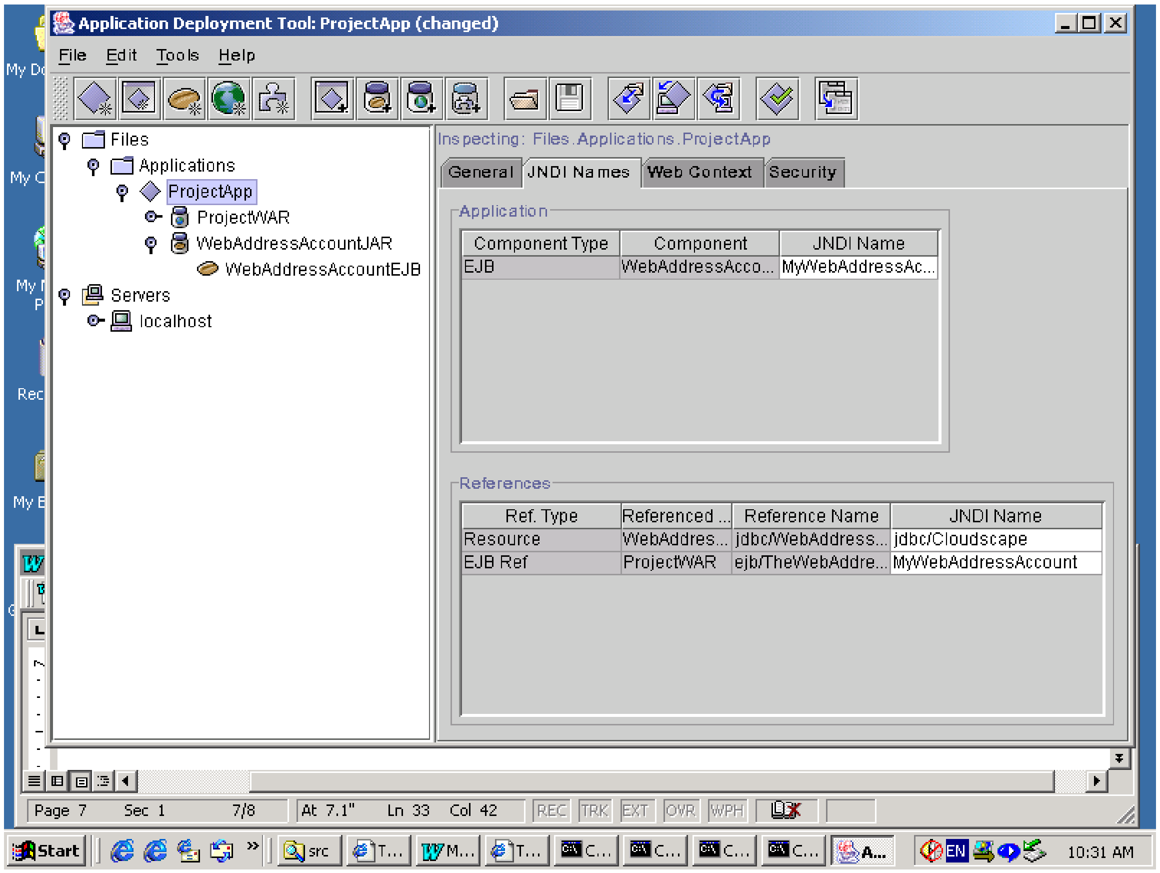
1. In the tree, select ProjectApp.
2. Select the JNDI Names tab.
3. To specify a JNDI name for the bean, in the Application table locate the WebAddressAccountEJB component and enter MyWebAddressAccount in the JNDI Name column.
4. To map the references, in the References table enter MyWebAddressAccount in the JNDI Name.

Figure-2 shows what the JNDI Names tab should look like after you've performed the preceding steps.

5. Deploying the J2EE Application
Now that the J2EE application contains the components, it is ready for deployment.

1. Select the ProjectApp application.
2. Select Tools->Deploy.
3. In the Introduction dialog box, confirm that ProjectApp is shown for the Object To Deploy and that localhost is shown for the Target Server.
4. Click Next.
5. In the JNDI Names dialog box, verify the names you entered in the previous section.
6. Click Next.
7. In the WAR Context Root dialog box, enter kwicProject in the Context Root field. When you run the Web client, the converter context root will be part of the URL.
8. Click Next.
9. In the Review dialog box, click Finish.

In the Deployment Progress dialog box, click OK when the deployment completes.



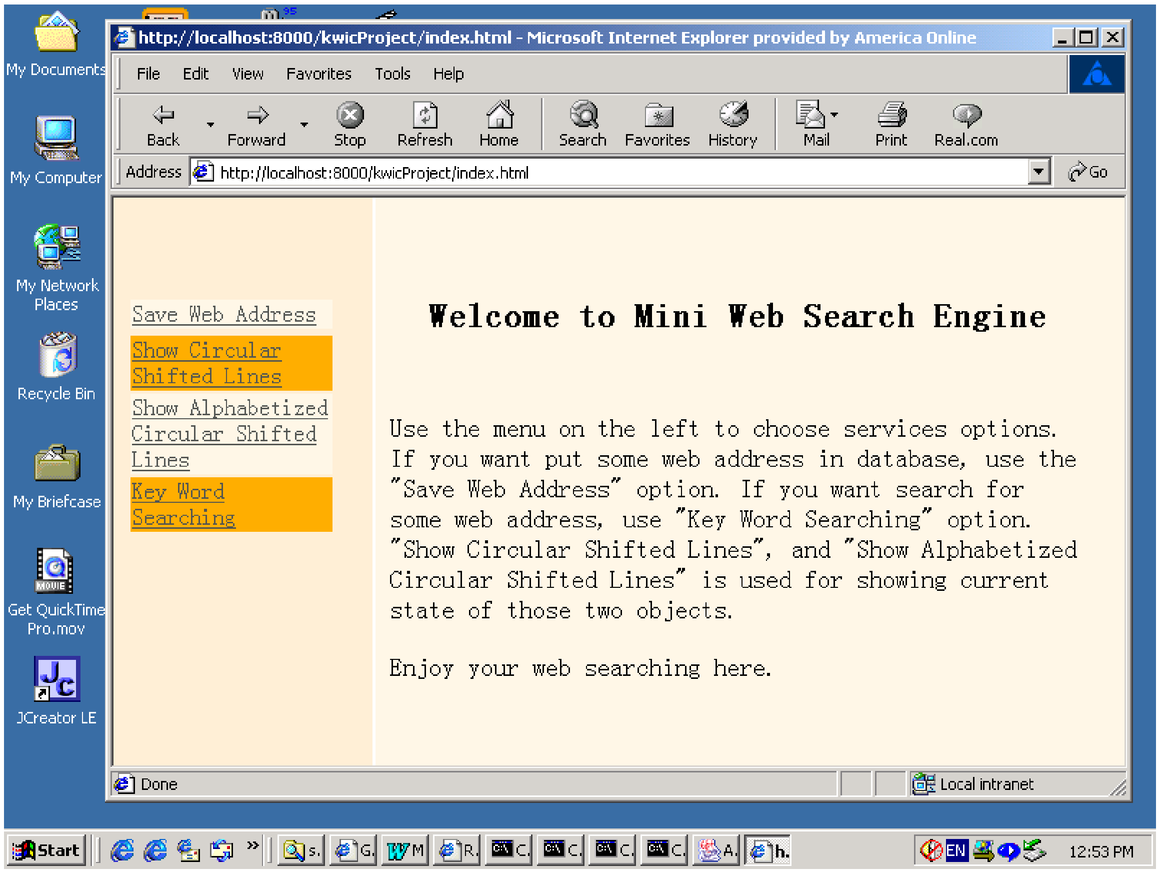
Component Type	Component	JNDI Name
EJB	WebAddressAcco...	MyWebAddressAc...

Ref. Type	Referenced ...	Reference Name	JNDI Name
Resource	WebAddres...	jdbc/WebAddress...	jdbc/Cloudscape
EJB Ref	ProjectWAR	ejb/TheWebAddre...	MyWebAddressAccount

Figure-2 WebAddressAccountApp JNDI Names

5. Running the Web Client

To run the Web client, point your browser at the following URL. Replace <host> with the name of the host running the J2EE server. If your browser is running on the same host as the J2EE server, you may replace <host> with localhost (http://localhost:8000/kwicProject).
http://<host>:8000/kwicProject
Figure-3


http://localhost:8000/kwicProject/index.html - Microsoft Internet Explorer provided by America Online
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Real.com
Address http://localhost:8000/kwicProject/index.html
Go
Save Web Address
Show Circular Shifted Lines
Show Alphabetized Circular Shifted Lines
Key Word Searching
Done
Local intranet
My Documents
My Computer
My Network Places
Recycle Bin
My Briefcase
Get QuickTime Pro.mov
JCreator LE
Start
12:53 PM

Welcome to Mini Web Search Engine

Use the menu on the left to choose services options. If you want put some web address in database, use the "Save Web Address" option. If you want search for some web address, use "Key Word Searching" option. "Show Circular Shifted Lines", and "Show Alphabetized Circular Shifted Lines" is used for showing current state of those two objects.

Enjoy your web searching here.