Designing Server Application

MUHAMMAD AJMAL P.

This program is a 3-tier server application developed in Java, which can be used to connect to three different types of databases such as FireBird, MySQL and MSAccess at a time. The screenshot of server application program output is shown in Fig. 1.

A database is an important and invaluable tool for any organisation, including electronics and IT industries. A database allows you to manage and use an incredible variety of information easily. But designing a database server application requires lot of understanding in the architectures and coding. Presented here is a client-server application which can be tested on a single PC or on multiple systems.

Fig. 1: Screenshot of program out in server

Features

This application can help you get started with Java RMI (remote method invocation), 3-tier application development, database application development, etc. Java RMI is nothing but a Java API that performs the object-oriented equivalent of remote procedure calls (RPC), with support for direct transfer of serialised Java objects and distributed garbage collection.

Tools required for development

1. Eclipse Indigo JEE
2. Java development tools (JDT) E3.7
3. Windows Builder Eclipse plug-in 1.5
4. Java JDK 1.6
5. Firebird 2.5
6. IBExpert

You also need 'ESoftDBExpress.jar' file along with these tools. This file includes all the database classes required for the application described here. 'ESoftDBExpress.jar' file is included in this month's EFY DVD.

There are five '.java' files in this project, where the 'ESoftServer.java' and the 'SettingsPanel.java' files are the main files. The 'MainMenu.java' file is for the menu and the 'Message.java' file is for loading external properties. The fifth is 'ESoftClient.java,' which should be in the client application side.

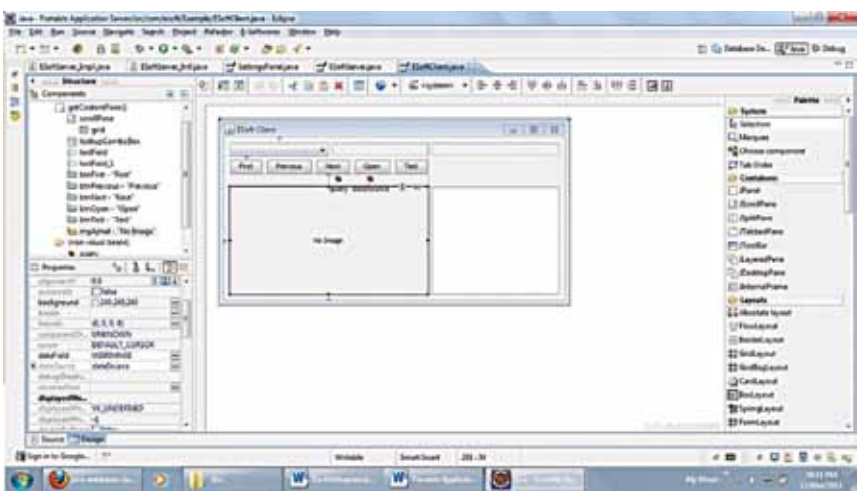

Fig. 2: Screen shot of Eclipse Windows Builder

Working. Practically, there should be at least two PCs to check the program, so that one works as server and the other as client. But in this project, you can check it on one PC by running the server as well as client sample applications simultaneously.

The RMI is initialised in the 'SettingsPanel' class constructor as given below.

```
try {
getFrame().
setRegistry(LocateRegistry.
createRegistry(Integer.parseInt(spr_
Port.getValue().toString().trim())));
((ESoftServer) frame).getRegistry().
bind(ESoftServer.SERVICE_NAME, get
Frame().getServer());

System.out.println("Server is running
. . . !");
getFrame().editorPane.
setText(getFrame().editorPane.
getText() + "\n" + "Server is running
. . . !");
} catch(final Exception e) {
// TODO Auto-generated catch block
((ESoftServer) frame).editorPane.
setText( ((ESoftServer) frame).
editorPane.getText() + "\n" +
e.getMessage());
e.printStackTrace();
}
```

By running this code, a new registry is created for RMI using the 'LocateRegistry.createRegistry(int port)' method. Then bind the 'ESoftServer_Intf' interface from 'ESoft-

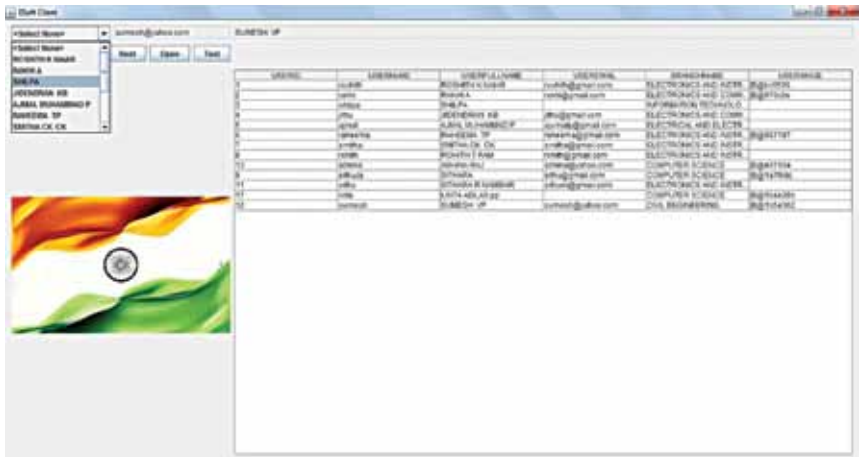

Fig. 3: Screenshot of sample client application (database)

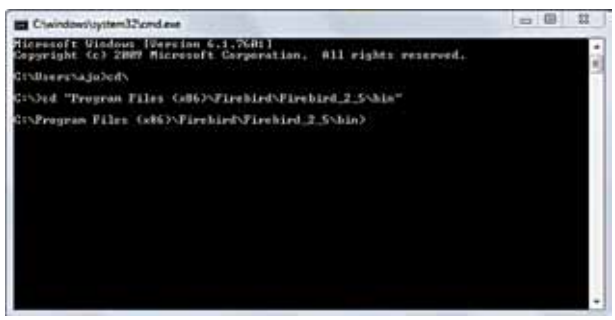

Fig. 4: Firebird path

Fig. 5: Client application testing window

Server.java' to the above registry in the name 'ESoftService.MobileServer' in 'ESoftClient.java' in the client application to point to these services in the corresponding port. Note that in larger projects, we need to write many services for different purposes. So we usually categorise them with a service name. Here we used 'ESoftService.MobileServer' as the service name.

First, we need to initialise the database for 3-tier access using the 'public void setConnection(final ESQLConnection connection, final DBConnectiondbc)' in the ESoftServer class. There are three ESQLQuerys and EDatasetProviders. Each one of them will be connected to the corresponding database when this method is called. Also, the corresponding provider is added to the 'ESoftServer_Impl' class of the ESoftDBExpress datasnap package, so that these dataset providers can be accessed at the client side.

Client sample application

A sample client application (ESoftClient.java) is developed and included in the 'com.esoft.Examples' package in this project. The client application is fully designed using Windows Builder plug-in of Eclipse (refer Fig. 2.)

The client application is connected to the server with the following code:

```
try {
final Registry registry = LocateRegistry.
getRegistry("localhost", 8092);
System.out.println(registry.list()
[0]);
serverIntf = (ESoftServer_Intf)
registry.lookup("ESoftService.
MobileServer");
serverIntf.Authenticate
("{123654789123654789}");
System.out.println(serverIntf.
getTestCode());
query.setCommandText("SELECT
USERNAME, USERFULLNAME, USERID,
USERIMAGE FROM USERMASTER");
} catch(final Exception e) {
        // TODO: handle exception
        e.printStackTrace();
}
```

To run both server and client in one system, 'localhost' is used. The port was set to 8092 in the server, so the same port is set in the client side. An authentication method is used in the 'ESoftDBExpressDatasnapESoftServer_Impl' class for security. The security code is set to '{123654789123654789}'. You can modify the code if you want.

Testing

After installing the required tools as mentioned earlier, run Eclipse.

1. Import the 'Portable Application Server.zip' file included in the EFY DVD into Eclipse. For that go to 'File->Import->General->Existing Project into Workspace' and browse the 'Portable Application Serve.zip' file.

You can just put the 'ESoftDBExpress.jar' file anywhere in the project folder, say, 'res' folder. Then refresh the project from Eclipse and add the 'ESoftDBExpresss.jar' file to project.

2. For this, from Eclipse, right click on the 'ESoftDBExpress.jar' file and go to 'Build Path' option, select 'Configure Build Path,' libraries and click 'Add JAR' file.

3. Firebird 2.5 is used for testing. Firebird is an open source software that can be downloaded from the Internet.

Firebird database 'ESoftECA.rar' file is included in EFY DVD. Install Firebird and restart your system.

4. Now you need to set the password for authentication. In Windows 7, go to 'Desktop→run→cmd→Firebird→bin>' directory as shown in screenshot Fig. 4. For Windows XP, go to 'C:\Program Files>Firebird\Firebird_2_5\bin>'

5. From there type 'Gsec -user sysdba –password masterke' and hit 'Enter' key. Then close the window.

6. Now got to project's 'src' folder from Eclipse workspace and open the 'messages.properties' file and set the password as 'masterke' in the code line 'FirebirdPassword=masterke'

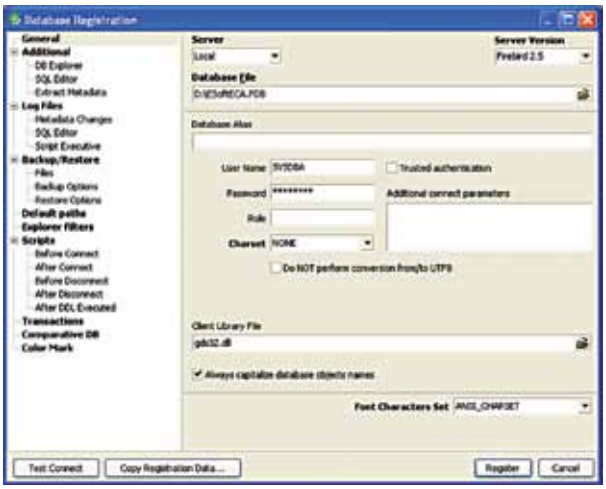

Fig. 6: IBExpert registration

7. Now open the 'ESoftClient.java' file in Eclipse workspace.

8. Go to code line 221 and enter the details as given below:

```
final Registry registry = LocateRegistry.
getRegistry("localhost", 8092);
```

Or search for localhost in that file. If you are running both client and server applications in the same computer then leave it as localhost itself. Otherwise, enter the IP address of the computer in which you want to access the information. In this example, we have to test both server and client on the same computer.

9. Then right click on 'Portable Application Server.' From workspace choose 'Run as→Run Configuration→ ESoftServer>' to run the server. If the server application is running fine, as shown in the screenshot Fig. 1, the project importing is successful.

There are only a few settings for this application:

(i) Check the 'Enable editing' check box

(ii) Select database type from combo box

(iii) Set the database path in the corresponding field

(iv) Set an open port [say 8092]

(v) Uncheck the 'Enable editing' check box

(vi) Click save button

Restart application

10. To run both the applications, open 'ESoftSever.java' click somewhere on the screen and hit F11 to run the server application. Then go to 'ESoftClient.java' and click somewhere in it and hit F11 to run client application.

11. Then click 'Test' button in the client application. You will get the message box 'Test Ok' as shown in Fig. 5.

12. Then click 'open' button. A data table along with picture will appear as shown in Fig. 2. Now you are done.

13. Using IBExpert, you can modify the tables and fields of the database. Note that IBExpert should be downloaded and installed in your system.

(i) Open IBExpert from desktop, click 'Menu→Database→Register Database' as shown in screenshot Fig. 6.

(ii) Click 'Register' database option. You will find the database registered in IBExpert. You can double click on the table USERMASTER to open that table and view and edit the details. ●

EFY Note

The source code of this project is included in this month's EFY DVD and is also available for free download on www.efymag.com website.

The author is a B.Tech in electronics and instrumentation. Presently, he is a software engineer at Software Associates, Calicut. His interests include software development, embedded systems and electronics circuit designing