

Getting the Most Out of Your Database:

Importing Oracle Java Database Connectivity
Drivers in NetBeans and Eclipse

Todd Wierzchowski
University of Wisconsin-Parkside
Department of Computer Science
November 23, 2011

Table of Contents

Sections

Introduction	2
NetBeans	3
Eclipse	7

Listing of Figures

NetBeans Figure 1: Determine the Library is missing	3
NetBeans Figure 2: Tools Menu	3
NetBeans Figure 3: Adding new Library, part 1	4
NetBeans Figure 4: Adding new Library, part 2	4
NetBeans Figure 5: Adding JAR to Library	5
NetBeans Figure 6: Checking Libraries folder	5
NetBeans Figure 7: Adding Library to Project	6
Eclipse Figure 1: Project > Properties	7
Eclipse Figure 2: Properties, Java Build Path, Libraries	7
Eclipse Figure 3: External jar added	8
Eclipse Figure 4: Checking Referenced Libraries	9

Introduction

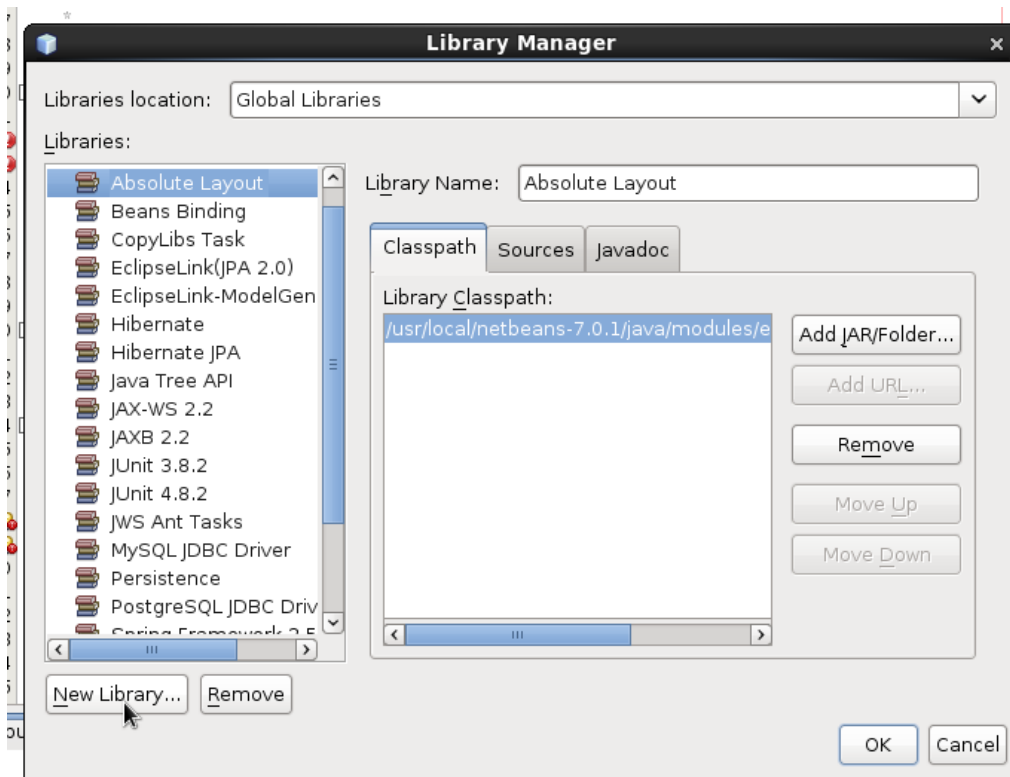
When using the NetBeans or Eclipse Integrated Development Environments (IDE), it is necessary to import the Oracle Java Database Connectivity (ojdbc) drivers in order for the Graphical User-Interface (GUI) you create to be able to connect properly to the Oracle database. The process of importing these drivers into either of the above listed IDEs is detailed below. Before beginning it is necessary to obtain a copy of the latest ojdbc jar file. At the time of writing it is currently ojdbc6.jar, which is up-to-date for use with Java version 1.6. For your convenience, this file is located on the Computer Science servers at: `/home/student/Classes/Cs380/ojdbc6.jar`

You should first copy this file to your home directory in a place where it will not move or be deleted. From the terminal run the following commands:

```
mkdir ~/ojdbc/
```

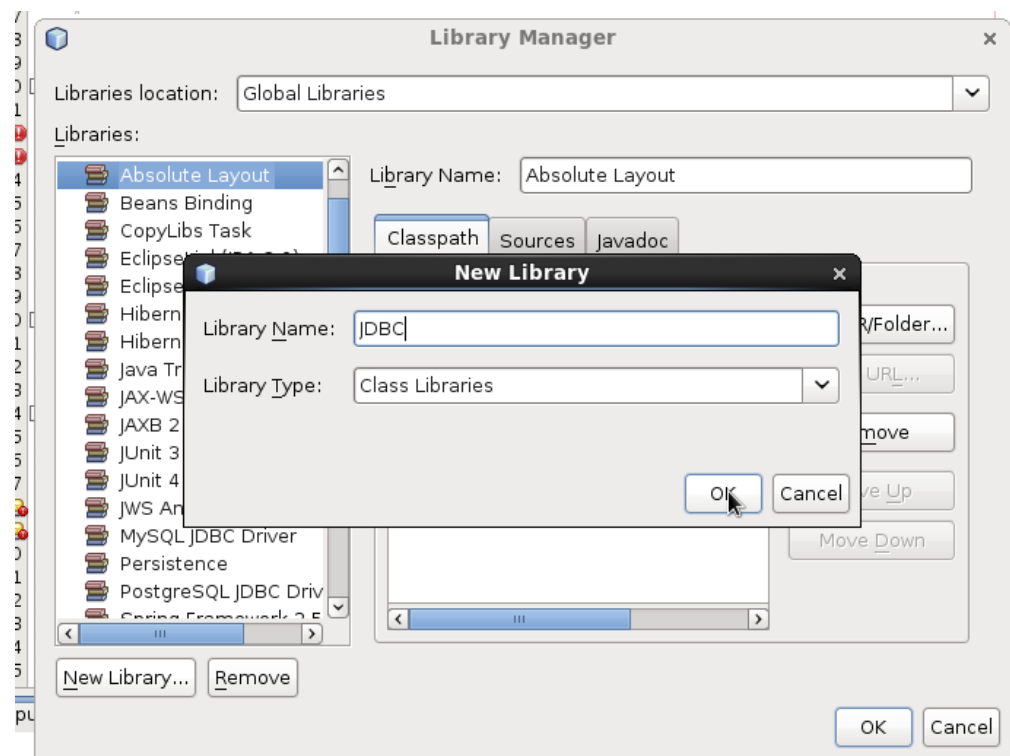
```
cp /home/student/Classes/Cs380/ojdbc6.jar ~/ojdbc/
```

These two commands will create a folder in your home directory called ojdbc, and copy the needed jar file into that directory. Once you have obtained the most up-to-date copy of the ojdbc jar file, the way you import into NetBeans or Eclipse varies slightly. Just follow the directions for your chosen IDE.



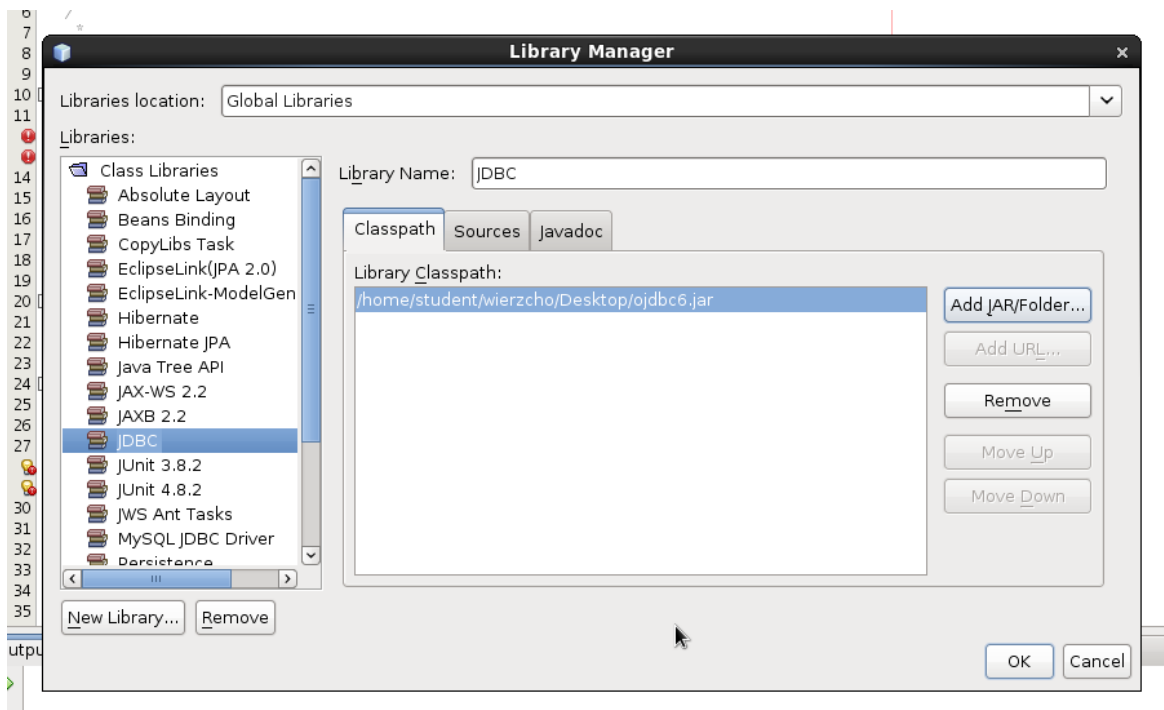
NetBeans Figure 3: Adding a new Library, part 1

Once the Library Manager is open, click on the New Library button.



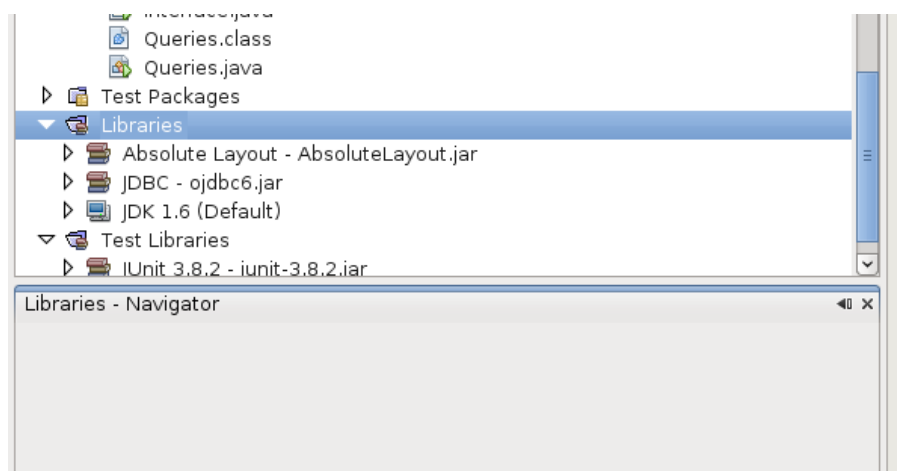
NetBeans Figure 4: Adding a new Library, part 2

Next, name your new Library, in this case it was simply named JDBC, and ensure it is of type Class Library. Click OK button.



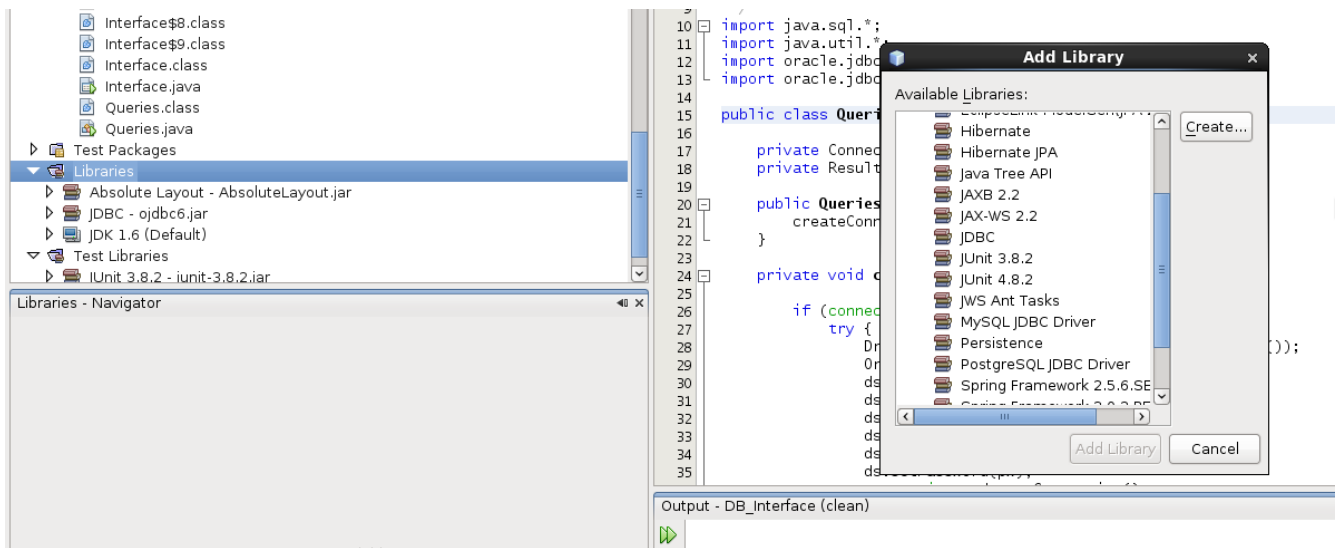
NetBeans Figure 5: Adding JAR to Library

With the new library created, make sure it is selected in the **Libraries** column, and the **Classpath** tab is selected, then click on the **Add JAR/Folder...** button. Navigate to the jar file you copied from the server located in the ojdbc folder in your home directory. Select the jar file and click the **Add JAR/Folder** button. Click the **OK** button to exit the Library Manager.



NetBeans Figure 6: Checking Libraries Folder

At this point it will have added the Library JDBC to the **Libraries** folder in your **Project Explorer**. The current project you have opened now has a reference to the ojdbc drivers and will work.



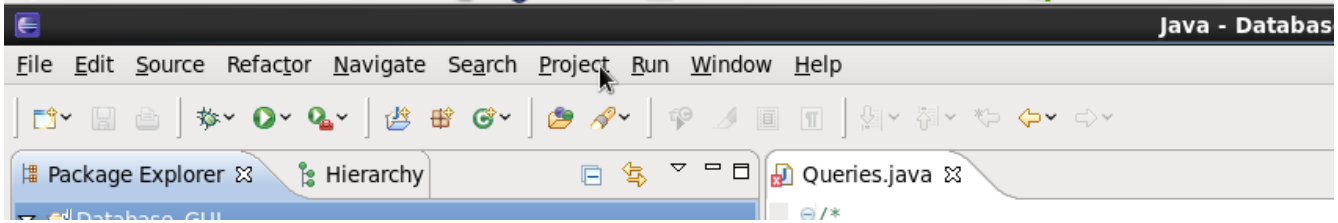
NetBeans Figure 7: Add Library to Project

Note: This only adds this library for the current project open in NetBeans.

If you need to add this library to another project right-click on the **Libraries** folder in your **Project Explorer** and select **Add Library**. Navigate to the Library you created with the ojdbc jar in it. Click the **Add Library** button and check to make sure it is listed under the **Libraries** folder in your **Project Explorer** on the left.

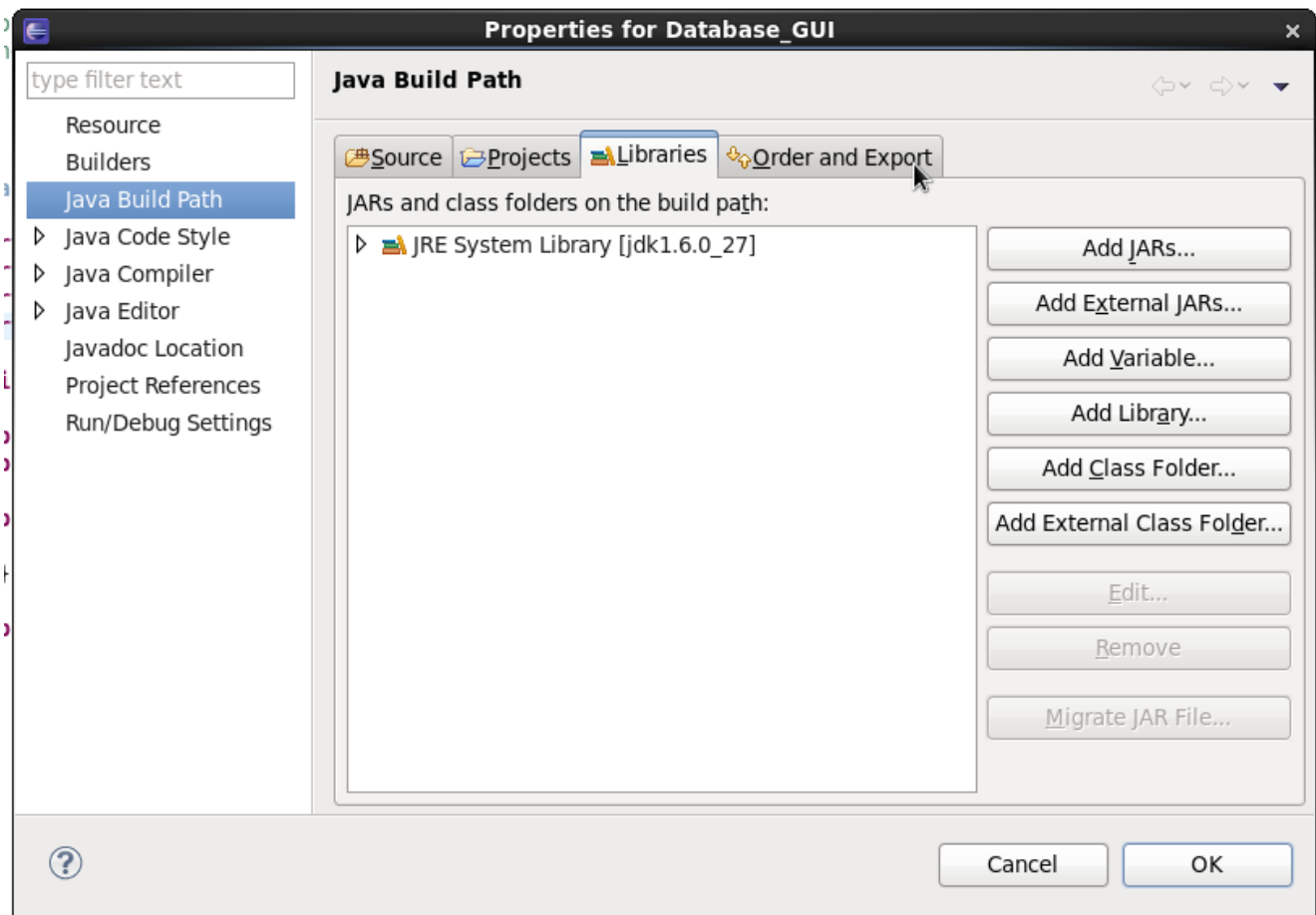
Eclipse

With your project open in Eclipse, a similar error message to the one given in NetBeans will be seen in Eclipse. The exact error message could not be captured in a screenshot, but it says “The import oracle can not be resolved”.



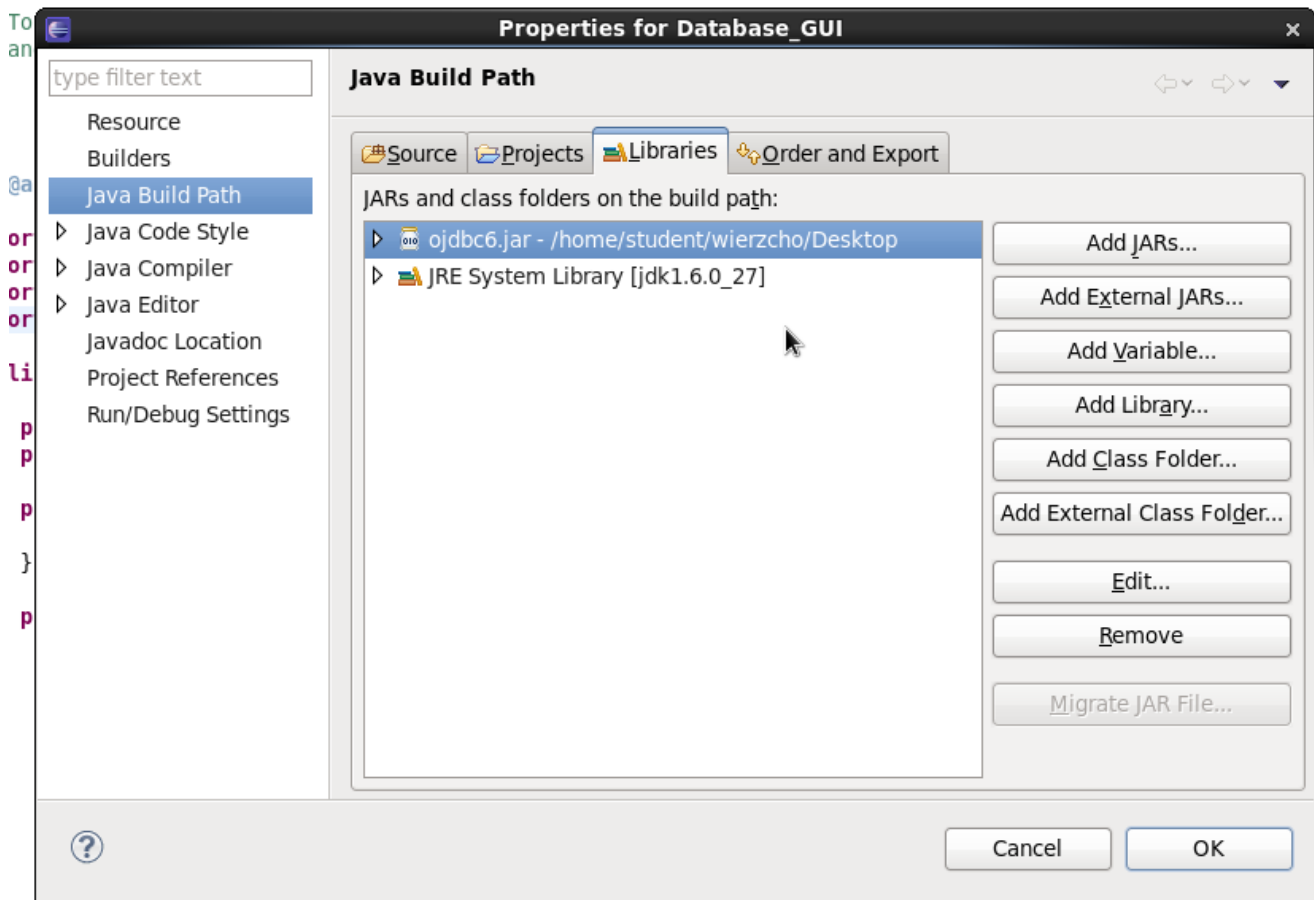
Eclipse Figure 1: Project > Properties

To resolve this conflict, select your project in the Package Explorer and open the Project menu at the top of the window, and click on the Properties option.



Eclipse Figure 2: Properties, Java Build Path, Libraries

In the properties window for the current project, select Java Build Path from the left column of options, then the Libraries tab. Select the Add External JARs... button and navigate to the ojdbc jar file. Click OK.

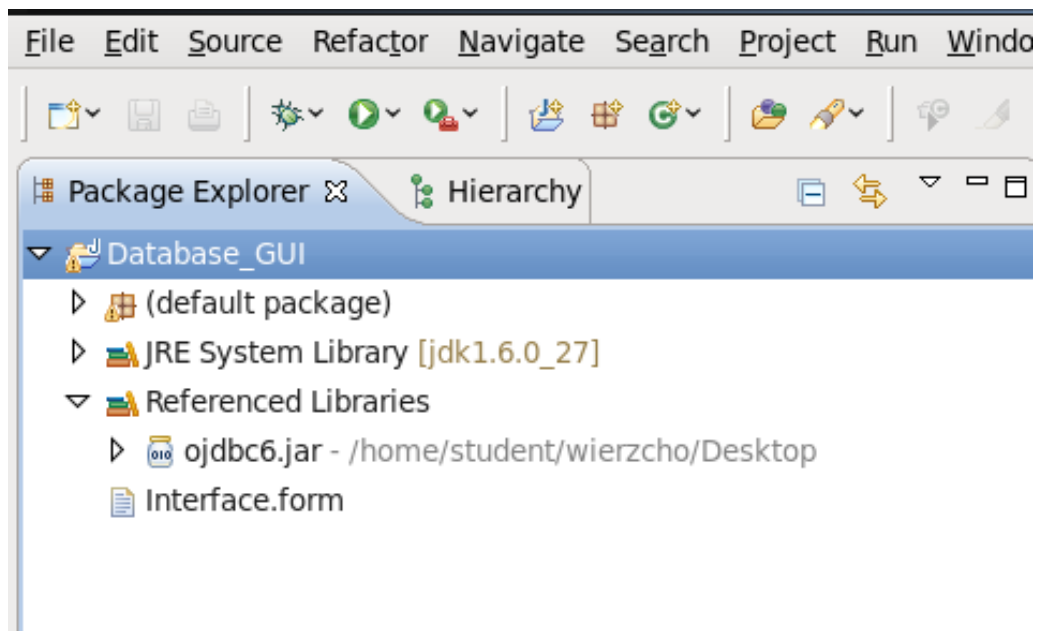


de setPassword(pwd);

Eclipse Figure 3: External jar added

If you followed the naming conventions given in the Introduction section, this should be in a folder called ojdbc located in your home directory. Click OK.

Note: In the above figure, the jar file was located on my Desktop instead of my home directory, hence the different in pathing to the jar.



Eclipse Figure 4: Checking Referenced Libraries

The final step is to check to make sure the link was made. In the **Package Explorer** on the left side of the screen, you should see a new tabbed drop-down option titled **Referenced Libraries**. If you expand this you should see the ojdbc jar file, along with the path to it.

Note: If you need this jar package in another project simply follow these steps again to reference the ojdbc drivers as part of the new project.