

A step by step guide to using JDBC with Eclipse

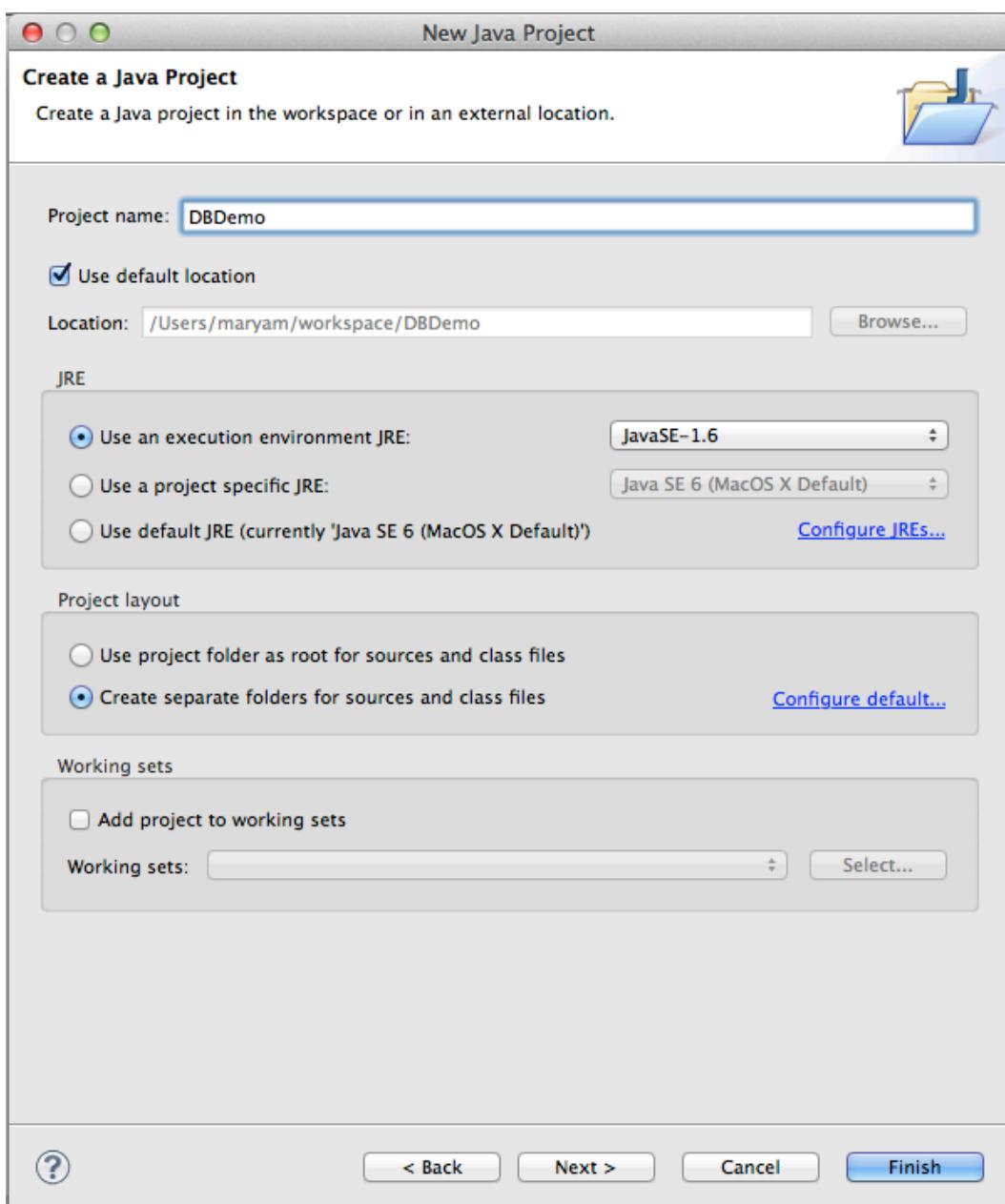
Step 1: Create an Eclipse Project

A project in Eclipse is essentially a folder containing all the source code and other files you need to build your program. Let's start by creating a project for your program.

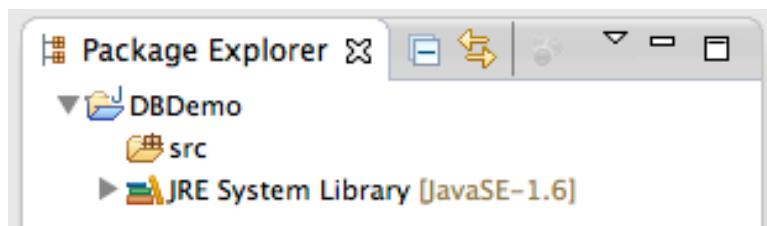
1. Open Eclipse
2. From the File menu select New → Java Project



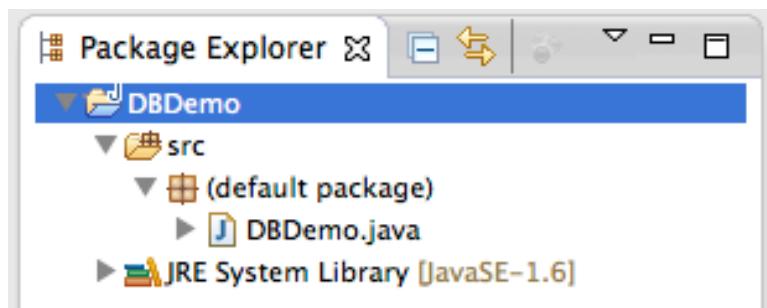
3. Give your project a name, like "DBDemo." You can use the default values for all the other settings. Make note of the project folder in the "Location" box; you will need to remember this later.



4. Press Finish to create the project
5. Eclipse might ask you whether you want to switch to the Java perspective. If so, say Yes.
6. You should see an empty project which looks something like this.



7. Download the JDBC starter file named DBDemo.java from the course web site at <http://www.ccs.neu.edu/home/kathleen/classes/cs3200/index.htm>
8. Copy DBDemo.java into the src folder in your project folder (from the Location box in step 3)
9. Right click on your project in Eclipse and select "Refresh." DBDemo.java should appear in the src folder.



10. Double click on DBDemo.java to open the file. You should take the time to look through the code and understand how it works later, but for now let's just get it running. This file has one major problem: it relies on a JDBC connection library which we haven't installed! That's the next step.

Step 2: Install the Connector/J JDBC driver

In this step, we install the Connector/J driver so our program can connect to MySQL.

1. Go to this web page: <http://www.mysql.com/downloads/connector/j/>
2. Press the download button next to the "Platform Independent (Architecture Independent), ZIP Archive" version. (Or the TAR version if you prefer; it doesn't really matter.)

Generally Available (GA) Releases

Connector/J 5.1.24

Select Platform:

Platform Independent

[Looking for previous GA versions?](#)

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-5.1.24.tar.gz)	5.1.24	3.5M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-5.1.24.zip)	5.1.24	3.8M	Download

MD5: cba8fbf00025e5f24086840c8bd60547 | [Signature](#)

MD5: f994df6a7c106eb8160d2e570ff2c853 | [Signature](#)

3. The next page will ask you to create an account. Instead, click the little blue link on the bottom that says “No thanks, just start my download.”

Begin Your Download - mysql-connector-java-5.1.24.zip

[Login Now or Sign Up for a free account.](#)

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system
- Comment in the MySQL Documentation

[Login »](#)

using my Oracle Web account

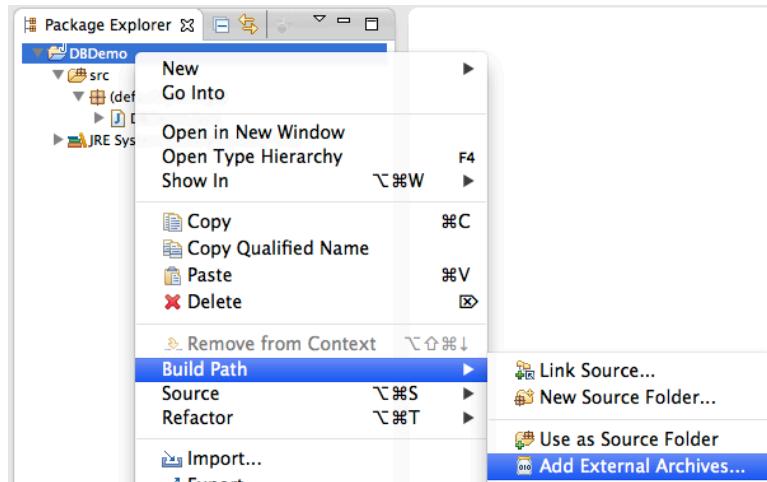
[Sign Up »](#)

for an Oracle Web account

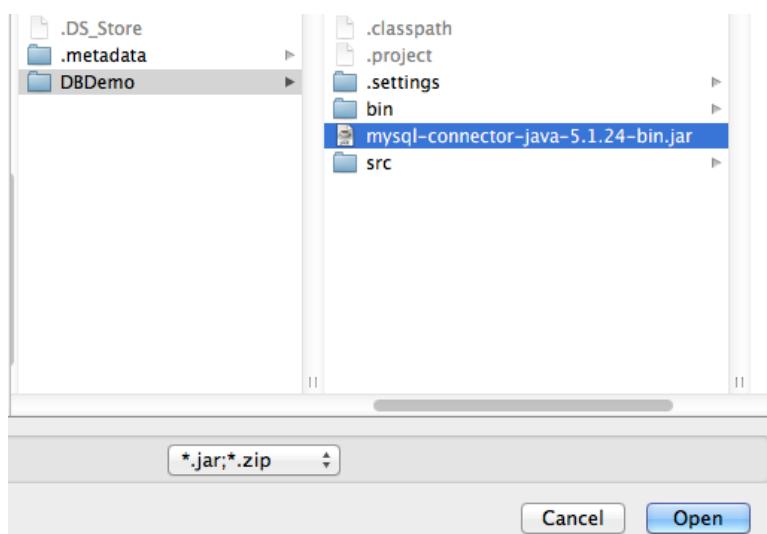
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

4. Open the .zip file you just downloaded, and look for a file inside named mysql-connector-java-5.1.24-bin.jar. Copy this file into your project directory (from the Location box in step 3 of the last section).
5. Now that we have the driver, we need to tell your project about it. Go back to Eclipse, right click on the project, and select Build Path → Add External Archives...



6. Select mysql-connector-java-5.1.24-bin.jar and press Open



7. Now we're ready to configure the project for your copy of MySQL and run it!

Step 3: Set up a simple database program

We have the program, we have the driver, and we just need a few final touches before we're ready to go.

1. Double click on DBDemo.java to open the file.
2. If you followed the MySQL tutorial's advice and created a user named root with the password root, then skip to the next step. If you have some other user name and password, look for the variables called `userName` and `password` and change them to the account you created.

```
public class DBDemo {

    /** The name of the MySQL account to use (or empty for anonymous) */
    private final String userName = "root";

    /** The password for the MySQL account (or empty for anonymous) */
    private final String password = "root";
```

3. Run the program by clicking the big green arrow button in the toolbar: 
4. Look in the Console in the bottom of the Eclipse window to see what happened. If everything worked, you should see something like this:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:
<terminated> DBDemo [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (Mar 17, 2013 6:55:41 PM)
Connected to database
Created a table
Dropped the table

5. If you saw that, you're done! If not, you have a problem to fix. Here are some things that can go wrong:

- MySQL might not be running
- Your user name and password might be wrong
- The credentials are right, but the user doesn't have permission to connect to the database

Try to get it working, if you can. There are some comments in DBDemo.java which might help out. If you can't get it to run, please don't hesitate to ask for help from the TAs! Good luck!