Java Database Connectivity (JDBC)

What is JDBC?

The Java Database Connectivity (JDBC) API is an industry standard methodology for defining how a client may access a database through the Java programming language. JDBC allows a Java program to execute statements in order to perform actions on a database. For this class's purposes, JDBC allows the Java program to connect to a SQLite database and handle any actions needed on the database through Java code.

Usage

In order to use JDBC with a SQLite database, a SQLite JDBC driver will be needed. This tutorial will use the one found here:

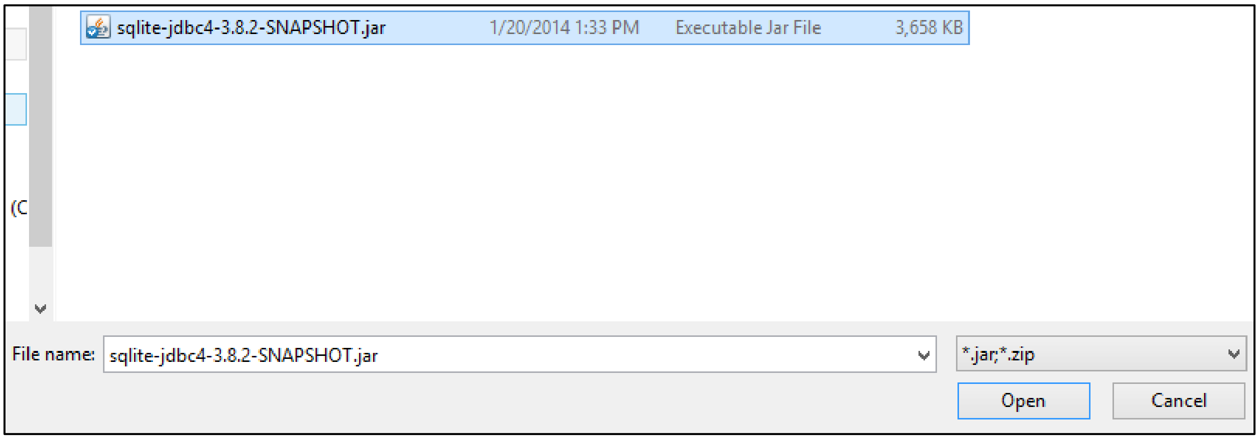
bitbucket.org/xerial/sqlite-jdbc/overview

The driver file that is downloaded is a JAR file that requires no configuration. Simply download the driver JAR file and when creating a new project in Eclipse:

1. Click the **Next** button on the Create a Java Project window
2. Click the **Libraries** tab, click the **Add External JARs…** button, and find the driver file

To add the JAR file to an existing project:

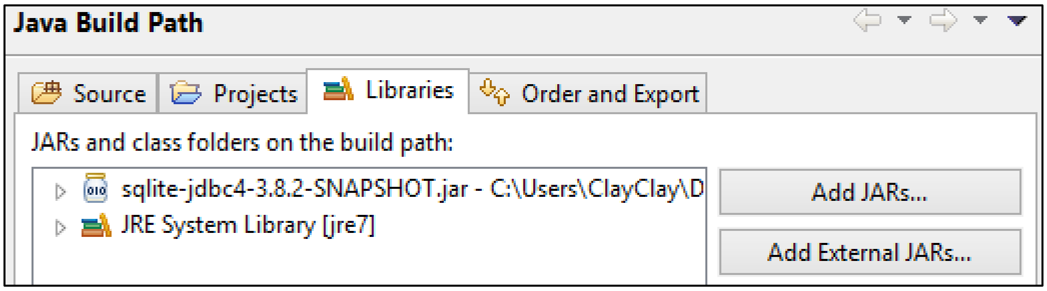
1. Right click the project in the Package Explorer and click **Preferences**
2. Click on **Java Build Path** and then continue from **Step 2** above.



sqlite-jdbc4-3.8.2-SNAPSHOT.jar 1/20/2014 1:33 PM Executable Jar File 3,658 KB

File name: sqlite-jdbc4-3.8.2-SNAPSHOT.jar

.jar;.zip



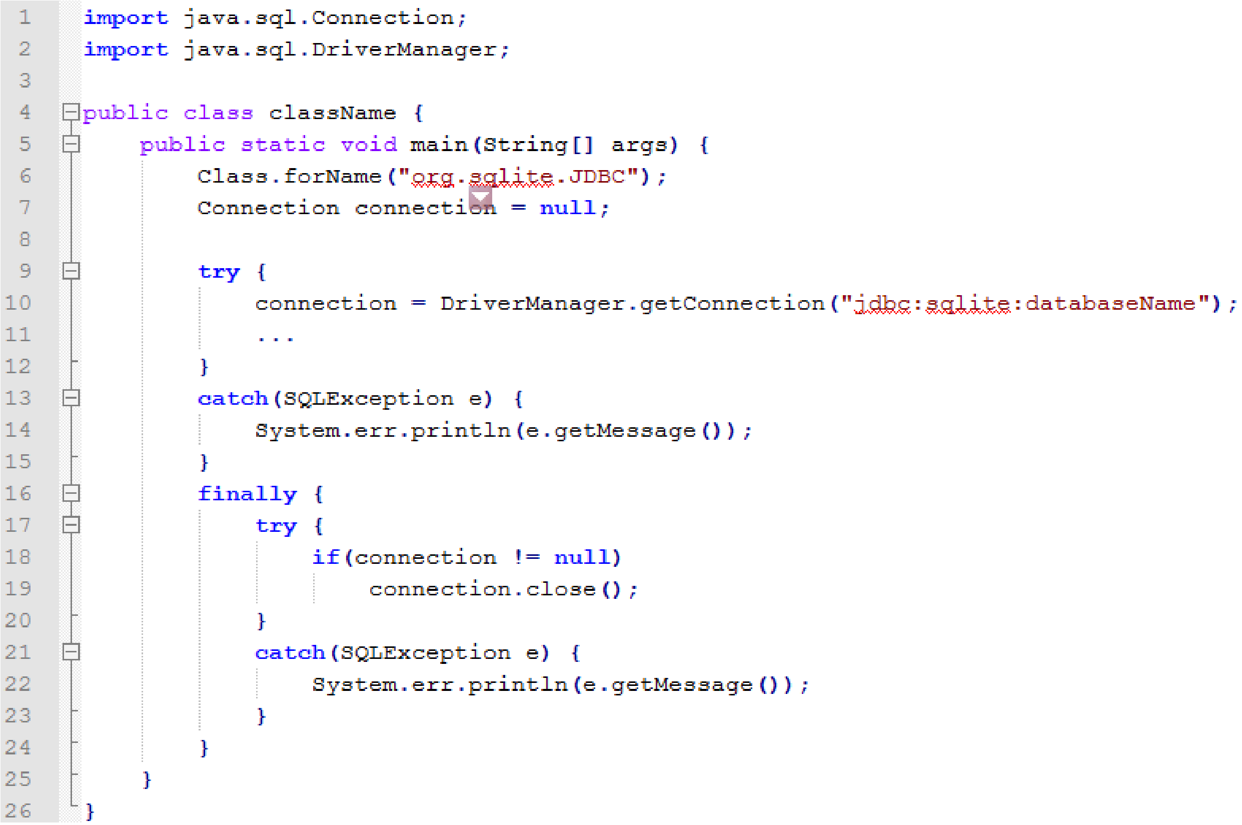
Java Build Path

JARs and class folders on the build path:

sqlite-jdbc4-3.8.2-SNAPSHOT.jar - C:\Users\ClayClay\D
JRE System Library [jre7]

Java Database Connectivity (JDBC)

Here is some example Java code that shows how to initialize a SQLite database connection:



```java
import java.sql.Connection;
import java.sql.DriverManager;

public class className {
    public static void main(String[] args) {
        Class.forName("org.sqlite.JDBC");
        Connection connection = null;

        try {
            connection = DriverManager.getConnection("jdbc:sqlite:databaseName");
            ...
        }
        catch(SQLException e) {
            System.err.println(e.getMessage());
        }
        finally {
            try {
                if(connection != null)
                    connection.close();
            }
            catch(SQLException e) {
                System.err.println(e.getMessage());
            }
        }
    }
}
```