The Oracle Installation bible

By Thia Kai Xin, Edited from JDBC installation by: Lichu Dang, Lixin Yu

The steps to enable your Java program to talk to Oracle database:

1. DL Oracle 11g (note got 2 files):
   http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html?ssSourceSiteId=ocomen

2. unzip both files into SAME folder

3. Install and TEST oracle following the steps carefully, create a new database oracle1:
   Link below has been removed by oracle
   http://www.oracle.com/technology/obe/11gr1_db/install/dbinst/windbinst2.htm

4. To setup our database

   - start > all programs > oracle -oradb11g_home1 > application development > sql developer

   - in sql developer: file > new >database connection

     1. connection name (aka any name u like): teamName

     2. username: system

     3. password: password1

     4. hostname: localhost

     5. port: 1521

     6. SID (aka database name): oracle1

        *if you receive an error saying network adapter cannot connect or not found, go start > all programs > oracle -oradb11g_home1 >configuration and mitigation tools> net configuration assistant> click yes all the way.

5. Add a new user before running the SQL script. In sql developer: right click “other user”, create user “user_admin” with password “password1”. Give user full role and privilege

6. Right click project1, disconnect and login as “user_admin”

7. Run the SQL script.

8. Add an admin account:

    - Click usertable>data tab>add user (+ sign) > username: admin password: fyprocks

9. Download the JDBC driver from Oracle at the following url.
   http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/htdocs/jdbc_112010.html

10. Download ojdbc6_g.jar

11. Put this ojdbc6_g.jar file to your Java classpath:

    - Oracle driver: place ojdbc6_g into C:\apache-tomcat-5.5.29\common\lib
    - right click **My Computer**,
    - choose **Properties**,
    - click **Advanced** tab,
    - click **Environment Variables** button,
    - choose the variable **CLASSPATH** in the User variables table,
    - click **Edit** button
    - in the variable value text field, append **;**
    - set classpath to C:\apache-tomcat-5.5.29\common\lib\ojdbc6_g.jar
    - if CLASSPATH is not present in the table, click **Add** button and type **CLASSPATH** in the variable name and type C:\apache-tomcat-5.5.29\common\lib in the variable value box.

12. Do a short simple test by issuing command in DOS or a shell:
    java oracle.jdbc.driver.OracleDriver
    You will get an error, but don't panic.

    If your error said:
    Exception in thread "main" java.lang.NoSuchMethodError: main
    Congradulation, you have setup JDBC driver correctly, despite what the error said.

    But if your error said:
    Exception in thread "main" java.lang.NoClassDefFoundError: oracle/jdbc/driver/OracleDrivera
    Your driver has not been setup correctly. Go back and review steps from 1 to 4.

13. Done JDBC setup.

14. Start to program and connect to Oracle through JDBC driver.
    Following is a simple test program to actually connect to Oracle and retrieve some data. (or download it here).

```
import java.sql.*;
import oracle.jdbc.driver.*;
public class DbTest {
public static void main (String args[]) throws Exception {

// Load the Oracle JDBC driver
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
// connect through driver
Connection conn = DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:uClaim"," claim_admin","oracle");
// Create Oracle DatabaseMetaData object
DatabaseMetaData meta = conn.getMetaData();
// gets driver info:
System.out.println("JDBC driver version is " + meta.getDriverVersion());
// Create a statement
Statement stmt = conn.createStatement();
// Do the SQL "Hello World" thing
ResultSet rset = stmt.executeQuery("SELECT TABLE_NAME FROM USER_TABLES");
while (rset.next())
System.out.println(rset.getString(1));
// close the result set, the statement and disconnect
rset.close();
stmt.close();
conn.close();
System.out.println("Your JDBC installation is correct.");
}
}
```

15. Compile and run the above DbTest. You should get the message **Your JDBC installation is correct**. If not, please see the following troubleshootings:

    - If you get an error message similar to this: *The system cannot find the file specified,*

      That means that you haven't told the JDBC where you store your file. Assume that the path to reach your file(s) is C:\cs157aProject\, you have to reset your CLASSPATH as the following:

C:\sjsu\cs157a\jdbc\classes12.zip;C:\cs157aProject
Refer to step 4 to reset your CLASSPATH. If you still have problem of compiling your program, restart your computer.

Note: Once you setup the CLASSPATH, all your java programs have to store under the same directory to compile, no matter you use JDBC or not.

- If you get an error message similar to this:

  Exception in thread "main" java.sql.SQLException: Io exception: Connection refused(DESCRIPTION=(TMP=)(VSNNUM=134238208)(ERR=12505)(ERROR_STACK=(ERROR=(CODE=12505)(EMFI=4))))
  That means your database name in the code is not correct. Change the ORCL in your connection string in the DbTest.java to the correct database name. Recall that you had created your database name when installing Oracle. If you don't remember it, you could search your Oracle installation directory for a file called **TNSNAMES.ORA** and open it. Find the SID value, that is your database name.

- If you get the error message:

  Exception in thread "main" java.sql.SQLException: ORA-01017: invalid username/pa ssword; logon denied

  That obviously means your user name or password is not correct. scott/tiger is the default login to Oracle database, but you may have changed it while in Oracle installation.

- If you get this error:

  Exception in thread "main" java.sql.SQLException: Io exception: The Network Adap ter could not establish the connection

  That hints your Oracle TNS listener service may not have been started or your port number is not correct. Go to Windows Services console and start the Oracle TNS listener service if it is not already started. And check your Oracle database listening port number.

16. Once the DbTest program runs correctly, you could start coding your own Java Database program. To make a good looking project, you should have knowledge of Java GUI programming such as Swing, AWT or Applet. Here we only get into details of the database programming portion.

- To connect through JDBC, the first thing you need is to load the JDBC driver to your program. Here is how you do it:
  DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
  This loaded driver enables you to make a connection to Oracle database.

- The next thing you would do is to make the actual connection:
  Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","system","oracle"); This tells the driver manager to create a connection to a database. The long parameter here is the essential part of connecting operation.
  1. The portion **jdbc:oracle:thin** instructs the driver manager to use the Oracle thin type jdbc driver, which is exactly the one we loaded in the previous step.
  2. The **@localhost** indicates the database host that you are connecting to. Here I used the local machine IP address. You could change it to a remote machine name such as **@sigma.mathcs.sjsu.edu**, if that is the Oracle instance you want to connect to.
  3. The number **1521** is the port number of the Oracle database. The default when Oracle is installed is to listen to port #1521, but it is configurable, so make sure your code is using the right number.
  4. The second and third parameters are the username and password for your Oracle database.

- Once the connection is made, you may want to start issuing your SQL statement to Oracle within your code. You may then create a java.sql.Statement object by writing:
  Statement stmt = conn.createStatement();

- Using the Statement object, you can then construct your SQL statement and execute it. The result of a SQL statement will be assigned to a ResultSet object.
  String sqlString = "SELECT * FROM SUPPLIER";
  ResultSet rset = stmt.executeQuery(sqlString);

- If your SQL statement runs successfully, you will be able to walk through the obtained ResultSet object to get result details. Here is one example:

```
6.      int numCols = 4; // there are 4 columns in SUPPLIER table
7.      while (rset.next()) { // walk through each row
8.        for (int i = 1; i<=numCols; i++) {
9.          System.out.println(rset.getString(i) + " "); // print each columen
10.       }
11.           System.out.println();
```

}

17. A complete database demo with GUI can be downloaded here:
    JdbcDemo.zip

---

for connection properties setup :
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@10.211.55.3:1521:uClaim
jdbc.user=user_admin
jdbc.password=password1

Oracle driver: place ojdbc6_g into C:\apache-tomcat-5.5.29\common\lib and set classpath to C:\apache-tomcat-5.5.29\common\lib\ojdbc6_g.jar

Amazing errors:
If your oracle DB suddenly stops working after a few weeks, it may be because your error log is full and oracle cannot start if your error log is full. So to solve this problem:

1. control panel>admin tools>event viewer.
2. Right click application> properties
3. Change max log size to 4992kb (or larger)
4. click clear log (don't bother to store the error messages)
5. Repeat for security and system.