Installing Java, Tomcat, and MySQL on Mac OS X 10.4.x

Ben Slade, 12/18/05
All feedback appreciated. Pls send feedback/questions to PublicMailbox at BenSlade dot com. Note, you'll be receive a challenge authentication email as an anti-spam measure.

Intro

These instructions are aimed at people who might not be familiar with the Unix environment underlying OS X. The instructions are therefore a little detailed in parts.

These instructions are for a Tomcat server running standalone. Ie., not for a Tomcat server running underneath an Apache webserver.

Installing Java

At the time of this writing, Mac OS X 10.4 comes with Java 1.4.2. These installation instructions refer to Java 1.5 so you must therefore download the Java 1.5 runtime from:

http://www.apple.com/support/downloads/

- Look for "Java 2 SE 5.0"
- Double-click the ".dmg" disk image file that you downloaded, and it will mount to appear as a disk in the Finder.
- Open the mounted disk, then double-click the Installer icon (.pkg file)
- Follow the onscreen instructions to complete the installation.

According to the release notes, this will add Java 1.5 to the system, but *not* make it the default version of Java. You must run the new "Java preferences utility" at `/Applications/Utilities/Java/J2SE 5.0/Java Preferences` to make Java 1.5 the default version on OS X. But this didn't seem to change the default version of Java everywhere on my system, so I had to do the following:

Start a terminal window (via Applications->Utilities->Terminal) and run the following commands (your Mac account must have administrator privilege to do this). Note, only type in the lines starting with the "$" sign, but do not type in the dollar sign:

```
$ cd /System/Library/Frameworks/JavaVM.framework/Versions
$ 'ls' -l CurrentJDK
   lrwxr-xr-x   1 root  wheel    5 May 11 16:36 CurrentJDK -> 1.4.2
$ sudo rm CurrentJDK
$ sudo ln -s 1.5.0 CurrentJDK
   lrwxr-xr-x   1 root  wheel    5 May 12 09:47 CurrentJDK -> 1.5.0
```

Setting Up the JAVA_HOME Environment Variable

On my system, I have the JAVA_HOME environment variable set to /Library/Java/Home which, via softlinks, eventually uses the above JDK dir. This is the way Apple's install packge sets things up.

To setup the JAVA_HOME variable on your system, if you're using the old tcsh shell for terminal commands, add "`setenv JAVA_HOME /Library/Java/Home`" to the ~/.tcshrc file (without the quotes). Else, if you're using the newer bash shell, add "`export JAVA_HOME=/Library/Java/Home`" to the file `~/.bash_profile` (without the quotes). If the `.bash_profile` file doesn't exist, go ahead and create it (note, you have to use "ls -a" to see hidden files). You should log out of your shell session and log back in for these changes to take effect.

After you see the CurrentJDK pointing to Java 1.5.0 you're good to go. To see if you're good type:

```
$ java -version
```

You should get the output:

```
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_02-56)
Java HotSpot(TM) Client VM (build 1.5.0_02-36, mixed mode, sharing)
```

For other Mac OS X Java related information see http://developer.apple.com/java/index.html

Installing MySQL

To download MySQL 4.1, go to the following web address:

http://dev.mysql.com/downloads/mysql/4.1.html

and look for "Mac OS X downloads", then look for "Installer package (Mac OS X v10.3)", then look for "Standard", then click on the "Pick a Mirror" link next to "Standard", then follow the menu prompts to download MySQL.

You should get a "disk image" file that has a name like "mysql-standard-4.1.11-apple-darwin7.8.0-powerpc.dmg". Double click on that dmg file and it will open up a folder window. To install MySQL, double click on the file in that folder named "mysql-standard-4.1.11-apple-darwin7.8.0-powerpc.pkg" and answer the prompts.

To setup MySQL so that it automatically starts up when your Mac boots, double click on the "MySQLStartupItem.pkg" and answer the prompts.

If you want to run MySQL commands from the Terminal command line, if you're using the tcsh shell (the default for pre OS X 10.3) add the following line to your the .tcshrc file in your login directory.

```
setenv PATH ${PATH}:/usr/local/mysql/bin
```

If you're using the bash shell (the default for OS X 10.3 and higher), add the following to the .bash_profile file in your login directory:

```
PATH=$PATH:/usr/local/mysql/bin
```

At this point, you should reboot your machine to make sure MySQL is working fine. After rebooting, you need to set passwords for the default accounts in MySQL (as per the Securing the Initial MySQL Accounts section of the MySQL documentation web pages):

Log into Mysql with the following command (don't type the "$"):

```
$ mysql -u root
```

Then run the following commands in MySQL (to remove anonymous MySQL logins) (commands in MySQL are not case sensitive):

```
DELETE FROM mysql.user WHERE User = '';
FLUSH PRIVILEGES;
```

Next you need to find out the name of the local Mac as determined by the MySQL installation software. To do this run the following command in MySQL :

```
select distinct(Host) as Host from mysql.user where Host != "localhost";
```

You'll get output back which looks like this:

```
+---------------------------------+
| Host                            |
+---------------------------------+
| My-Computer.local               |
+---------------------------------+
```

In this example `Benjamin-Slades-Computer.local` is the name of the local computer.

Now run the following commands to set the MySQL root password:

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
SET PASSWORD FOR 'root'@'My-Computer.local' = PASSWORD('newpwd');
```

Where you replace newpwd with your own root password and replace `Benjamin-Slades-Computer.local` with the name from from the select distinct query above. I think the password is upper/lowercase sensitive.

Setup the a Test Login in MySQL

Logged into mysql as root, do the following where 'prod_password' is the password for the production login account prod_login:

```
create database javatest;
grant all on javatest.* to 'test_mysql_login'@'localhost' identified by 'test_mysql_password';
flush privileges;
```

Where obviously, the 'test_password' is replaced with your own password . Don't use the root password (created above) for this. Also, don't change the database name "javatest" because it's a default name used below.

Setting Up Tomcat

Tomcat is software that permits a web request to be directed to a piece of Java code (servlet) with some associated infrastruture to handle cookies and request/reply coordination. These functions are collectively referred to as a "web container". Tomcat used to also be referred to as Catalina. The Catalina name is still used in some of the config setup below.

Tomcat can be downloaded from Tomcat Downloads at jakarta.apache.org (note, tis page currently seems to take you to a broken download page for Tomcat 5.5. Use this web page instead for Tomcat 5.5). You only need the "Core" download (skip the "Deployer" and "Admin" downloads).

For the instructions below, you'll need the `gnutar` utility to extract the .tar.gz files. This utility used to be available under the "Mac OS X Developer Tools" web page at Apple, but marketing people have changed that to the Tools XCode web page. You have to download the whole (~750mb as of 11/05) XCode integrated development environment in order to get gnutar *and* you first have to signup for an Apple Developer Connection (ADC) account to do so. You can probably find other ways to get gnutar on the Internet, but a quick search didn't turn anything up for me. To make this easier, I left an OS X version of `guntar` (ver 1.13.25) on my website at benslade.com/misc/gnutar.zip (from the Fink distribution, compiled under OS X 10.3.x, runs under 10.4.x)

Here are the instructions for installing Tomcat. Instructions starting with a "$" mean run the command (not including the "$"):

Download the 5.x version of Tomcat. For me the filename was apache-tomcat-5.5.12.tar.gz

```
$ sudo mv apache-tomcat-5.5.12.tar.gz /usr/local
$ cd /usr/local/
```

Create the directory /usr/local/apache-tomcat-5.5.12:

```
$ gnutar -xzvf apache-tomcat-5.5.12.tar.gz
$ sudo rm apache-tomcat-5.5.12.tar.gz
```

The following permits you to use `/usr/local/tomcat` to access tomcat files:

```
$ sudo ln -s apache-tomcat-5.5.12 tomcat
```

Change the ownership of the Tomcat files to "myusername" (replace with your login username):

```
$ sudo chown -R myusername: apache-tomcat-5.5.12
```

Setup the CATALINA_HOME environment variable to point to `/usr/local/tomcat` for your shell. See Setting Up the JAVA_HOME Environment Variable above.

Create bin directory and make it your current directory:

$ cd; mkdir bin; cd bin

Now you can create a file called `start_tomcat` with the following contents (this assumes the JAVA_HOME environment variable was defined above):

```
#!/bin/sh
export CATALINA_HOME=/usr/local/tomcat
$CATALINA_HOME/bin/startup.sh
```

And create another file called stop_tomcat with the following contents:

```
export CATALINA_HOME=/usr/local/tomcat

    $CATALINA_HOME/bin/shutdown.sh
```

Make the two scripts you just created executable:

```
$ chmod u+x *_tomcat
```

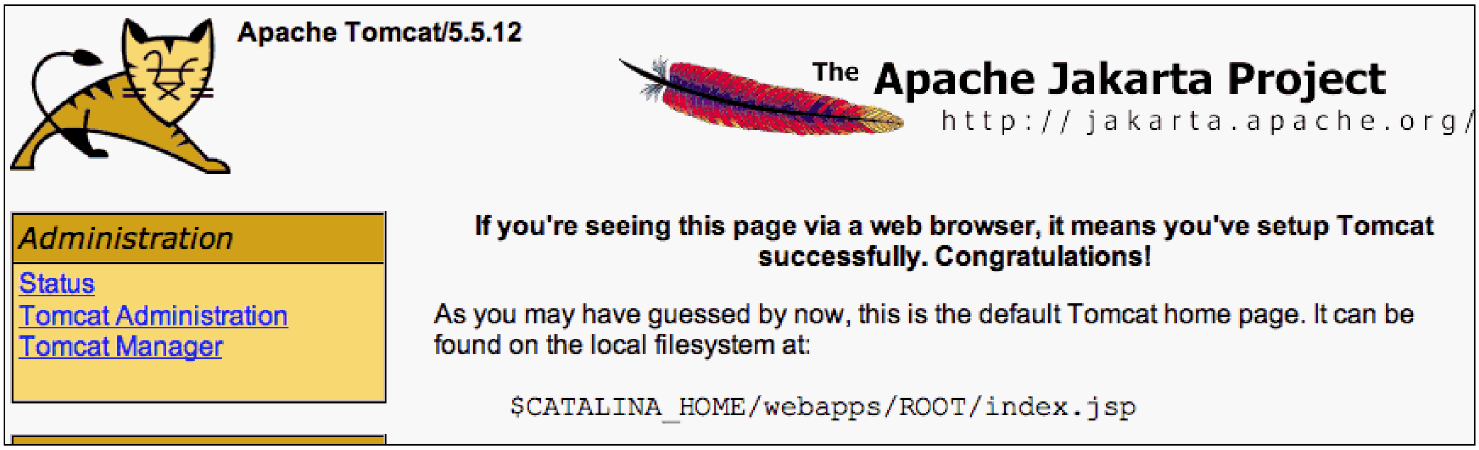
Add /Users/*myusername*/bin to your PATH environment variable, where *myusername* is replaced with your login name. If you're using the csh or tcsh shell add the following to the ~/.tcshrc file:

```
setenv PATH ${PATH}:/Users/myusername/bin
```

If you're using the Bash shell, add the following to your ~/.bash_profile file:

```
export PATH=${PATH}:/Users/myusername/bin
```

You can now use start_tomcat and start_tomcat to start and stop the tomcat server. Sometimes it takes Tomcat a little while to startup, say 10-20 seconds. Use the "top" shell command (terminal command) watch Tomcat CPU usage during startup. When usage dies down to near zero, it's done starting up. If tomcat is running the web page http://localhost:8080 should display:



If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

As mentioned on the Tomcat welcome page, you should setup usernames and passwords in the file `$CATALINA_HOME/conf/tomcat-users.xml`. My entry looks like this (with italics replaced appropriately):

```
<user username="myusername" password="mypassword" roles="tomcat,admin,manager"/>
```

You have to stop and restart Tomcat after adding users and/or changing passwords in Tomcat.

General documentation for Tomcat can be found here. To look for errors to go $CATALINA_HOME/logs. I find most database access errors seem to show up in the localhost*.log file.

Testing a JSP in Tomcat

Create a directory under `$CATALINA_HOME/webapps` called something like `MyApps`. `MyApps` is referred to as the "web application" directory. Copy the file `HelloWorld.jsp` into the `MyApps` directory. In your web browser go to the URL `http://localhost:8080/MyApps/HelloWorld.jsp` and see if you get a web page saying "Hello World" with the date and time.

Setting Up the Database Interface Drivers SQL Connectors Between Tomcat and MySQL

After you've setup the Tomcat server and MySQL, you need to setup software that connects the two.

Download the most recent "Connector/J" from http://dev.mysql.com/downloads/connector/. General documentation for this can be found at: http://dev.mysql.com/doc/refman/5.0/en/java-connector.html. Tomcat specific installation doc's can be found at Using Connector/J with Tomcat .

Again you need to extract the files in the download using the gnutar command (regular gunzip and OS X's tar won't work. See above for getting gnutar). Place extracted directory under /usr/local and create a generic softlink to it like so:

```
$ sudo ln -s mysql-connector-java-3.1.11 mysql-connector-java
```

Where *mysql-connector-java-3.1.11* is replaced with whatever version you downloaded.

Next place a link to the mysql-connector jar file in your Tomcat `$CATALINA_HOME/common/lib` directory (the "ln -s" below is all one line through the ".jar"):

```
$ cd $CATALINA_HOME/common/lib
$ sudo ln -s /usr/local/mysql-connector-java-3.1.11/mysql-connector-java-3.1.11-bin.jar mysql-
connector-java-3.1.11-bin.jar
```

Next you need to create the file `$CATALINA_HOME/webapps/`*`MyApps`*`/META-INF/context.xml` with the following content (substituting for *test_mysql_login* and *test_mysql_password* appropriately):

```
<Context  reloadable="true" >

  <Resource name="jdbc/TestDB" auth="Container" type="javax.sql.DataSource"
               maxActive="100" maxIdle="30" maxWait="10000"
               username="test_mysql_login" password="test_mysql_password"
               driverClassName="com.mysql.jdbc.Driver"
               url="jdbc:mysql://localhost:3306/javatest?autoReconnect=true"/>
</Context>
```

Note that if you do work in another web application directory other than MyApps, you'll need to create a META-INF/context.xml file in that web application directory too.

Test the Connection Between Tomcat and MySQL

Copy the file TestTomcatToMySQLConn.jsp to `$CATALINA_HOME/webapps/MyApps/TestTomcatToMySQLConn.jsp` and then invoke the URL:

http://localhost:8080/MyApps/TestTomcatToMySQLConn.jsp

You should get a message like "Hello World from MySQL".

On some Tomcat installations, you might have to restart the Tomcat server after creating/changing the META-INF/context.xml or `TestTomcatToMySQLConn.jsp` files. My Tomcat 5.5 server seemed to notice changes automatically.

To Do's

I need to figure out how to specify a database other than "javatest" for the default login database.