

## Setup Instructions for DQA Development

DQA is easiest to develop in Eclipse. But this is not required. Here are the instructions to the standard setup:

- Get Source Forge account  
The source code is stored in Source Forge. You should register an account and then send an email to Nathan Bunker with your new Source Forge id to request to become a member of the project.
- Install Maven  
Maven is a build process that is way awesome. It does a lot of what ant does but even more. Installing is as easy as ant and makes the rest of the dev process so much better.
  - Follow instructions here: <http://maven.apache.org/download.html>
- Install Eclipse  
Development does not require Eclipse, but it's what the developers on this project are using and so you may find it easiest to use this.
- Install Maven Eclipse plugin  
This plugin supports functions that help with using both Maven and Eclipse. You install this from within Eclipse. Go to Help >> Install software and use the second link below to get the software.
  - Instructions: <http://www.eclipse.org/m2e/>
  - Install link for Eclipse installer: <http://download.eclipse.org/technology/m2e/releases>
- Install SVN plugin  
This supports connecting to SVN. Eclipse doesn't support SVN out of the box. Install this just like the Maven Eclipse plugin.
  - Instructions: <http://subclipse.tigris.org/>
  - Eclipse link: [http://subclipse.tigris.org/update\\_1.8.x](http://subclipse.tigris.org/update_1.8.x)
- Install Oracle jar into Maven  
All the other jars in this project are open source, but you will need to add the Oracle jar manually. If you do not plan to use Oracle then you will need to comment out the oracle dependency in the pom.xml file instead.
  - Modify the script below to indicate the location of your Oracle jar and then run:  
`mvn install:install-file -Dfile={ORACLE_HOME}/jdbc/lib/ojdbc6.jar  
-Dpackaging=jar -DgroupId=com.oracle -DartifactId=oracle -Dversion=10.2.0`
- Setup database
  - Oracle: Better scripts are coming soon, contact Nathan Bunker for now
  - MySQL: Better scripts are coming soon, contact Nathan Bunker for now
  - Hypersonic: This is ready to go and you don't need to do anything to run Hypersonic
- Check out code
  - In Eclipse switch to SVN Repository perspective.
  - Add new SVN Repository at this URL: <https://ois-dqa.svn.sourceforge.net/svnroot/ois-dqa> You will have to use your source forge credentials
  - Open new folder, then brows to trunk folder and check out dqa-wicket.

- Choose "New Project Wizard"
  - On the wizard give the project a name and then finish
- Setting up project so it compiles
  - On the command line, go to the project root folder. (Same folder the pom.xml is in.)
  - Prep the file for Eclipse by running: `mvn eclipse:eclipse`
  - Go back to Eclipse, right-click on project and choose refresh.
  - The project should now have no compile errors and be setup correctly.
- Setup database parameters.
  - In Eclipse go to `src/main/resources` and open `hibernate.cfg.xml`.
  - Eclipse may try to be cute and show you the XML doc in a special window, you can select the Source tab on the bottom and switch to source.
  - By default the file is configured to work with HSQL. Change this to your preferred database by copying the lines of code from either the `hibernate.cfg.oracle.xml` or `hibernate.cfg.mysql.xml` file. As you will see these files are only a snippet of the full file so be sure to copy and replace carefully.
  - Change the connection parameters so the url/username/password is correct. For Hypersonic there is no need to make this change.
- Make sure database is running
- Running the project
  - To run, go into Test folder and find the Start java file.
  - Right click on Start and Run As >> Java Application.
  - Now the application will start in Jetty and the output will show in the Console inside Eclipse
  - Remember to stop it using the red stop button if you want to stop and restart it
- Building the project
  - On the command line, in the project root folder. (Same folder the pom.xml is in.) run this command: `mvn install`
  - Maven will first compile, then run the JUnit tests (database has to be running so unit tests will work), and then build the war.
  - The war file is in the target directory and is named with the current version.
  - You will probably want to rename the war file to simply `dqa.war`
- Deploy
  - This can be deployed on Tomcat, Glassfish, or it may work on your favorite J2EE app server. Try it!