Setup Instructions for OIS Matching Development

OIS Matching 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
- Check out code
  - In Eclipse switch to SVN Repository perspective.
  - Add new SVN Repository at this URL: https://svn.code.sf.net/p/oismatching/code/trunk 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
- Install SecondString jar into Maven
  Project requires SecondString to run. This needs to be installed into the Maven repository.
  - The Second String project page is here: http://secondstring.sourceforge.net/
  - Download the second string jar
  - Modify the script below to load the jar into maven:
    ```
    mvn install:install-file -Dfile=secondstring-20060615.jar -Dpackaging=jar\
    -DgroupId=com.wcohen -DartifactId=ss -Dversion=20060615
    ```
- 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.
- 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
    - Code is now running at http://localhost:8286 Here are the entry points:
        - /MatchPatientServlet - Allows for comparing two patients, normally this is accessed from other servlets
        - /RandomServlet – Creates a random set of patients, one set that matches and one that doesn't.
        - /GenerateWeightsServlet - Starts the genetic algorithm. Refresh to display status.
        - /RandomScriptServlet – Creates a list of random patients for testing.
        - /TestMatchingServlet – Allows for testing current match settings against a test script
    - 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 runing so unit tests will work), and then buld 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, our it may work on your favorite J2EE app server. Try it!