

Importing eagle-i code into Eclipse

The eagle-i source code is a multi-module maven project stored in a Subversion source control repository. This guide outlines a procedure for setting up the Eclipse IDE in order to import the eagle-i code. We assume a Mac running Mountain Lion, but much will be similar in other platforms. This is not meant to be an exhaustive guide, but rather to give you an idea of how to set up an environment - different OSs, different versions of Eclipse, different installed plugins, etc. introduce subtle differences that would be hard for us to cover here.

We have tested this procedure with Eclipse version (Kepler).

Set up Eclipse

Download or use your favorite Eclipse version and Edition (plain Java or J2EE both work fine), and make sure you are using a **Java 7** JDK/JRE.

m2eclipse

m2eclipse (m2e) provides reasonable integration of a maven project in Eclipse. Luckily, the m2e plugin is part of the standard Eclipse bundle since the Indigo release. If you have an older version of Eclipse (Helios or older), you'll have to install the plugin manually (it is not clear if that is supported any more - please consult the interwebs).

Subclipse

Whatever your Eclipse version, you'll need to make sure you have a Subversion plugin - either Subclipse or Subversive. We prefer Subclipse - to install it, follow the instructions here: [Subclipse installation](#). We use version 1.8 (later versions don't have the SvnKit option yet, which is what works in MacOSX). Once installed (requires a restart of Eclipse), go to `Preferences->Team->SVN` and select SvnKit as client interface.

m2eclipse Subclipse connector

You will also need to add the Subclipse connector for m2e, in order to checkout the code directly as a maven project. To do so, go to `Preferences -> Maven -> Discovery`, click on `Open Catalog` scroll until you see the connector m2e-subclipse (in the Team Providers section) and install it.

Check out the eagle-i code

Open the SVN perspective and add the eagle-i svn repository: <https://open.med.harvard.edu/svn/eagle-i-dev>

Open the repository, then the apps folder, then right click on `trunk` and select `Checkout as Maven project...` You can accept the defaults or customize as you please (e.g. add a prefix to all checked-out projects, add them to a filtered list, etc.). If you're prompted to select additional m2e connectors, accept the suggestions and continue. After a little while you'll see Eclipse building the workspace. A full build will take quite some time. Note that this initial build is not a maven compilation, but an Eclipse compilation. At the end of all this, you will have an eclipse project per maven module.

If your Eclipse edition supports Javascript, you will notice error markers in some javascript files. These are third party libraries (such as JQuery) - go figure. You may safely ignore them, the project will build fine (if they annoy you, you can turn off Javascript validation in Eclipse's preferences).

To build the entire maven project, right click on the root module `eagle-i-base` and select `run as -> maven install`. The built artifacts will reside in the `target` directory of each module/project.

There is a chance that some tests may fail with a `java.lang.OutOfMemoryError` when doing this full build. If this happens open the Run Configurations in Eclipse for this build. Go to the JRE tab and in the section VM arguments add the following:

```
-DargLine="-Xmx2048m -XX:MaxPermSize=1024m"
```