# Importing SHRINE code into Eclipse

The SHRINE 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 SHRINE code.  We assume a Unix-like system, like Linux or OS X, 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.

## Set up Eclipse

Download or use your favorite Eclipse version and Edition (plain Java or J2EE both work fine).

### 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.

### 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.

### ScalaIDE

All SHRINE modules except for one are written in Scala. We recommend installing the ScalaIDE plugin. Be sure to install a version for your Eclipse version and for Scala 2.10.x; this update site installs a version suitable for Eclipse Kepler and Scala 2.10.x, for example.

ScalaIDE downloads
ScalaIDE documentation

### (Optional) Groovy support

One SHRINE module is written in Groovy. Get the Groovy plugin for Eclipse here.

## Check out the SHRINE code

Open the SVN perspective and add the SHRINE svn repository: https://open.med.harvard.edu/svn/shrine

Open the repository, 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.

To build the entire maven project, right click on the root module `shrine-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"
```