

How to build Naked Objects from Source

Before You Start	2
Prerequisite Software	2
Check out the Source Code	2
Jimi Dependency	2
Build Maven Plugins.....	3
Build the Parent POM	4
Building the Full Distribution	5
Building just the Maven-Released Modules (the core framework + plugins)	6
Building in Eclipse	7
Hints and Tips.....	8

If you are thinking about modifying or contribute to Naked Objects, then you'll want to be able to build Naked Objects from source.

Naked Objects is built using Maven 2, so (provided you are on the Internet to download any dependencies) it pretty much builds itself. Full details of Maven modules are at http://nakedobjects.org/wiki/Maven_Modules.

Although you can just use Maven from the command line, you'll almost certainly want to use an IDE for proper development. We use Eclipse IDE with the M2Eclipse plugin. This generates the .project and .classpath files on-the-fly from the Maven pom.xml; that's why these files aren't checked into source code. NetBeans 6.7+ and IntelliJ also have built-in support for Maven (we haven't used either of these in anger, though).

Naked Objects is hosted on SourceForge at <https://nakedobjects.svn.sourceforge.net/svnroot/nakedobjects>.

How to build Naked Objects from Source

Before You Start

Prerequisite Software

Install a Subversion client, for example TortoiseSVN (<http://tortoisesvn.tigris.org>).

Install Java 5, setup JAVA_HOME

Install Maven 2.0.9 or later, setup MAVEN_HOME, add *mvn* to PATH.

Optional, but recommended: Install Eclipse 3.5 (JEE edition).

Also recommended is:

- the M2Eclipse plugin (<http://m2eclipse.codehaus.org/>).
- Subclipse plugin (<http://subclipse.tigris.org/>)

Check out the Source Code

The framework is at [../framework/trunk](#). In addition, there are also some Maven plugins for preparing the documentation. These reside in [../developers/maven-plugins/trunk](#).

Check out both directories.

Jimi Dependency

To build the full distribution (including documentation), you'll first need to install the jimi.jar file (not available in the central repository due to licensing issues) into your local repository:

- Download the jimi jar file. You'll find it within the JimiProClasses.zip downloadable from <http://java.sun.com/products/jimi/>.
- Install into your local Maven repository using:

```
mvn install:install-file -D groupId=com.java -D artifactId=jimi \
-D version=1.0 -D packaging=jar -D file=/path/to/jimi.jar
```

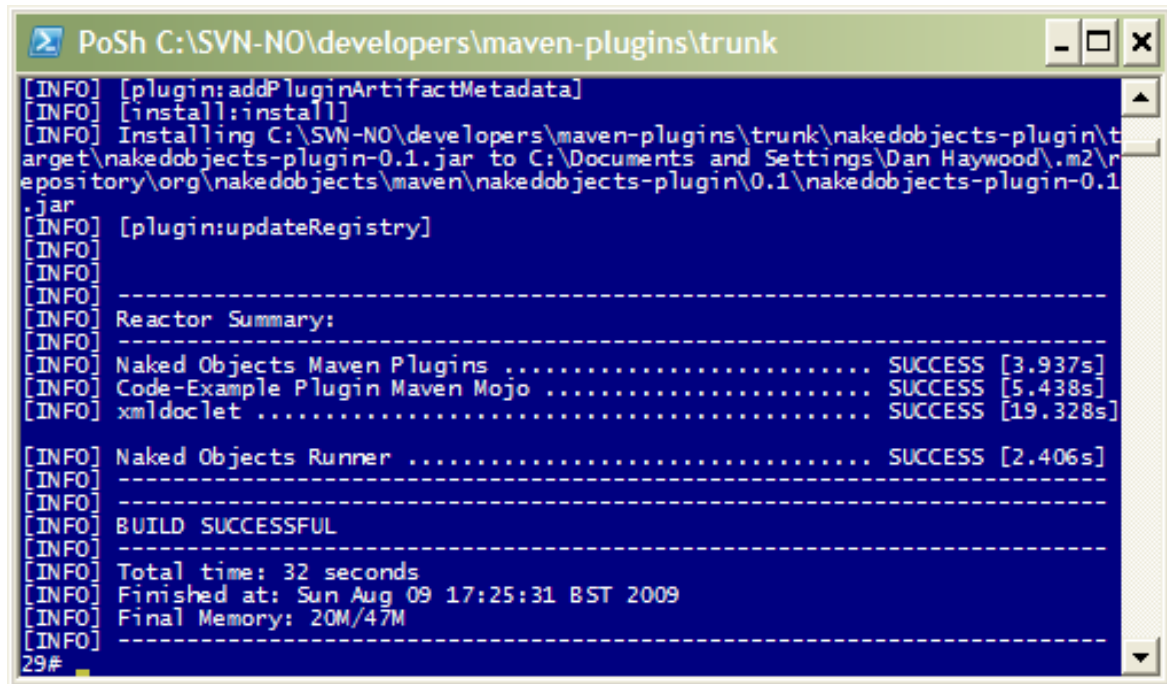
How to build Naked Objects from Source

Build Maven Plugins

The Naked Objects documentation uses a number of Maven plugins specific to NO itself.

```
$ cd ~/developers/maven-plugins/trunk
$ mvn clean install
```

You should end up with:



```
PoSh C:\SVN-NO\developers\maven-plugins\trunk
[INFO] [plugin:addPluginArtifactMetadata]
[INFO] [install:install]
[INFO] Installing C:\SVN-NO\developers\maven-plugins\trunk\nakedobjects-plugin\t
arget\nakedobjects-plugin-0.1.jar to C:\Documents and Settings\Dan Haywood\m2\re
pository\org\nakedobjects\maven\nakedobjects-plugin\0.1\nakedobjects-plugin-0.1
.jar
[INFO] [plugin:updateRegistry]
[INFO]
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Naked Objects Maven Plugins ..... SUCCESS [3.937s]
[INFO] Code-Example Plugin Maven Mojo ..... SUCCESS [5.438s]
[INFO] xmldoclet ..... SUCCESS [19.328s]
[INFO]
[INFO] Naked Objects Runner ..... SUCCESS [2.406s]
[INFO]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO]
[INFO] Total time: 32 seconds
[INFO] Finished at: Sun Aug 09 17:25:31 BST 2009
[INFO] Final Memory: 20M/47M
[INFO]
29#
```

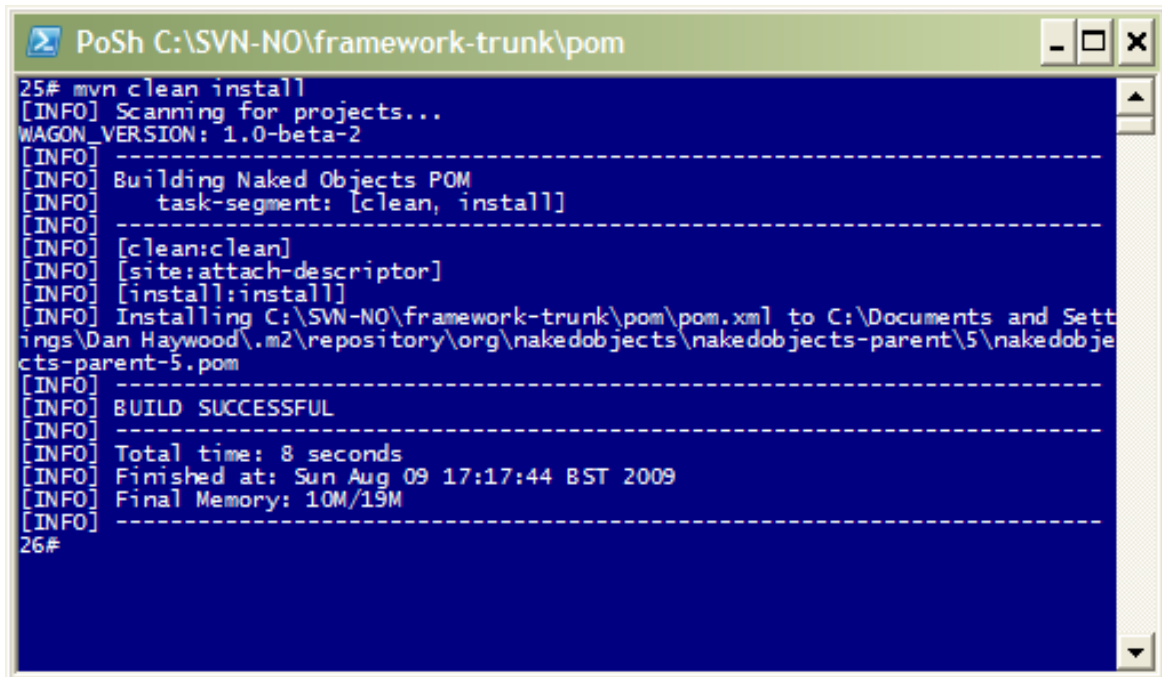
How to build Naked Objects from Source

Build the Parent POM

Build the parent (“corporate”) POM . (NB: the version of the parent POM doesn’t match that of the framework, this is by design).

```
$ cd ~/framework/trunk
$ cd pom
$ mvn clean install
```

You should end up with:



```

PoSh C:\SVN-NO\framework-trunk\pom
25# mvn clean install
[INFO] Scanning for projects...
WAGON_VERSION: 1.0-beta-2
[INFO] -----
[INFO] Building Naked Objects POM
[INFO] task-segment: [clean, install]
[INFO] -----
[INFO] [clean:clean]
[INFO] [site:attach-descriptor]
[INFO] [install:install]
[INFO] Installing C:\SVN-NO\framework-trunk\pom\pom.xml to C:\Documents and Settings\Dan Haywood\.m2\repository\org\nakedobjects\nakedobjects-parent\5\nakedobjects-parent-5.pom
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 8 seconds
[INFO] Finished at: Sun Aug 09 17:17:44 BST 2009
[INFO] Final Memory: 10M/19M
[INFO] -----
26#
```

How to build Naked Objects from Source

Building the Full Distribution

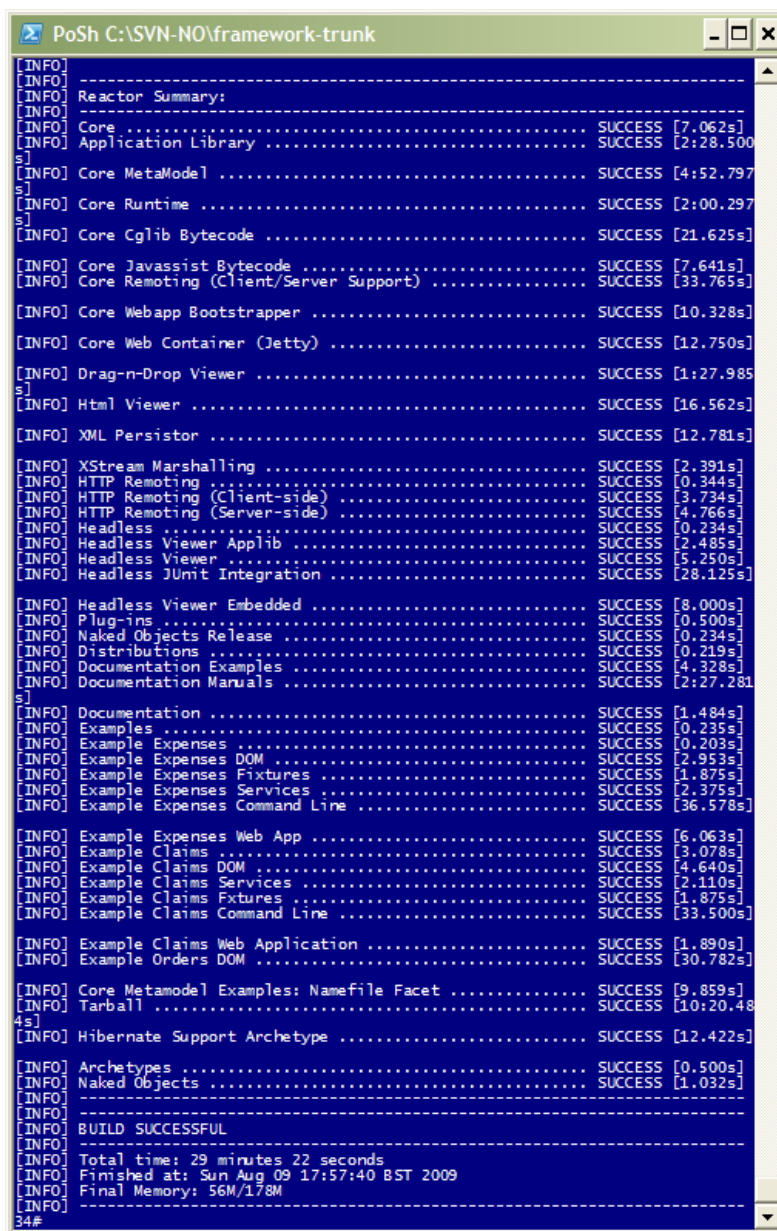
The complete build will build the Maven-released modules (the core framework + plugins), the documentation, examples and a couple of tarballs downloadable from sourceforge:

- one of the tarballs is for Maven users, and includes just the documentation and supporting files (such as icons and templates)
- the other tarball is for non-Maven users, and includes all the libraries and dependencies.

To build the full distribution, use:

```
$ mvn clean install -P all
```

You should end up with:



```
[INFO] -----
[INFO] Reactor Summary:
[INFO] Core ..... SUCCESS [7.062s]
[INFO] Application Library ..... SUCCESS [2:28.500s]
[INFO] Core MetaModel ..... SUCCESS [4:52.797s]
[INFO] Core Runtime ..... SUCCESS [2:00.297s]
[INFO] Core Cglib Bytecode ..... SUCCESS [21.625s]
[INFO] Core Javassist Bytecode ..... SUCCESS [7.641s]
[INFO] Core Remoting (Client/Server Support) ..... SUCCESS [33.765s]
[INFO] Core Webapp Bootstrapper ..... SUCCESS [10.328s]
[INFO] Core Web Container (Jetty) ..... SUCCESS [12.750s]
[INFO] Drag-n-Drop Viewer ..... SUCCESS [1:27.985s]
[INFO] Html Viewer ..... SUCCESS [16.562s]
[INFO] XML Persistor ..... SUCCESS [12.781s]
[INFO] XStream Marshalling ..... SUCCESS [2.391s]
[INFO] HTTP Remoting ..... SUCCESS [0.344s]
[INFO] HTTP Remoting (Client-side) ..... SUCCESS [3.734s]
[INFO] HTTP Remoting (Server-side) ..... SUCCESS [4.766s]
[INFO] Headless Viewer Applib ..... SUCCESS [0.236s]
[INFO] Headless Viewer ..... SUCCESS [2.485s]
[INFO] Headless JUnit Integration ..... SUCCESS [5.250s]
[INFO] Headless JUnit Integration ..... SUCCESS [28.125s]
[INFO] Headless Viewer Embedded ..... SUCCESS [8.000s]
[INFO] Plugins ..... SUCCESS [0.500s]
[INFO] Naked Objects Release ..... SUCCESS [0.234s]
[INFO] Distributions ..... SUCCESS [0.219s]
[INFO] Documentation Examples ..... SUCCESS [4.328s]
[INFO] Documentation Manuals ..... SUCCESS [2:27.281s]
[INFO] Documentation ..... SUCCESS [1.484s]
[INFO] Examples ..... SUCCESS [0.235s]
[INFO] Example Expenses ..... SUCCESS [0.203s]
[INFO] Example Expenses DOM ..... SUCCESS [2.953s]
[INFO] Example Expenses Fixtures ..... SUCCESS [1.875s]
[INFO] Example Expenses Services ..... SUCCESS [2.375s]
[INFO] Example Expenses Command Line ..... SUCCESS [36.578s]
[INFO] Example Expenses Web App ..... SUCCESS [6.063s]
[INFO] Example Claims ..... SUCCESS [3.078s]
[INFO] Example Claims DOM ..... SUCCESS [4.640s]
[INFO] Example Claims Services ..... SUCCESS [2.110s]
[INFO] Example Claims Fxtures ..... SUCCESS [1.875s]
[INFO] Example Claims Command Line ..... SUCCESS [33.500s]
[INFO] Example Claims Web Application ..... SUCCESS [1.890s]
[INFO] Example Orders DOM ..... SUCCESS [30.782s]
[INFO] Core Metamodel Examples: Namefile Facet ..... SUCCESS [9.859s]
[INFO] Tarball ..... SUCCESS [10:20.484s]
[INFO] Hibernate Support Archetype ..... SUCCESS [12.422s]
[INFO] Archetypes ..... SUCCESS [0.500s]
[INFO] Naked Objects ..... SUCCESS [1.032s]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 29 minutes 22 seconds
[INFO] Finished at: Sun Aug 09 17:57:40 BST 2009
[INFO] Final Memory: 56M/178M
[INFO] -----
34#
```

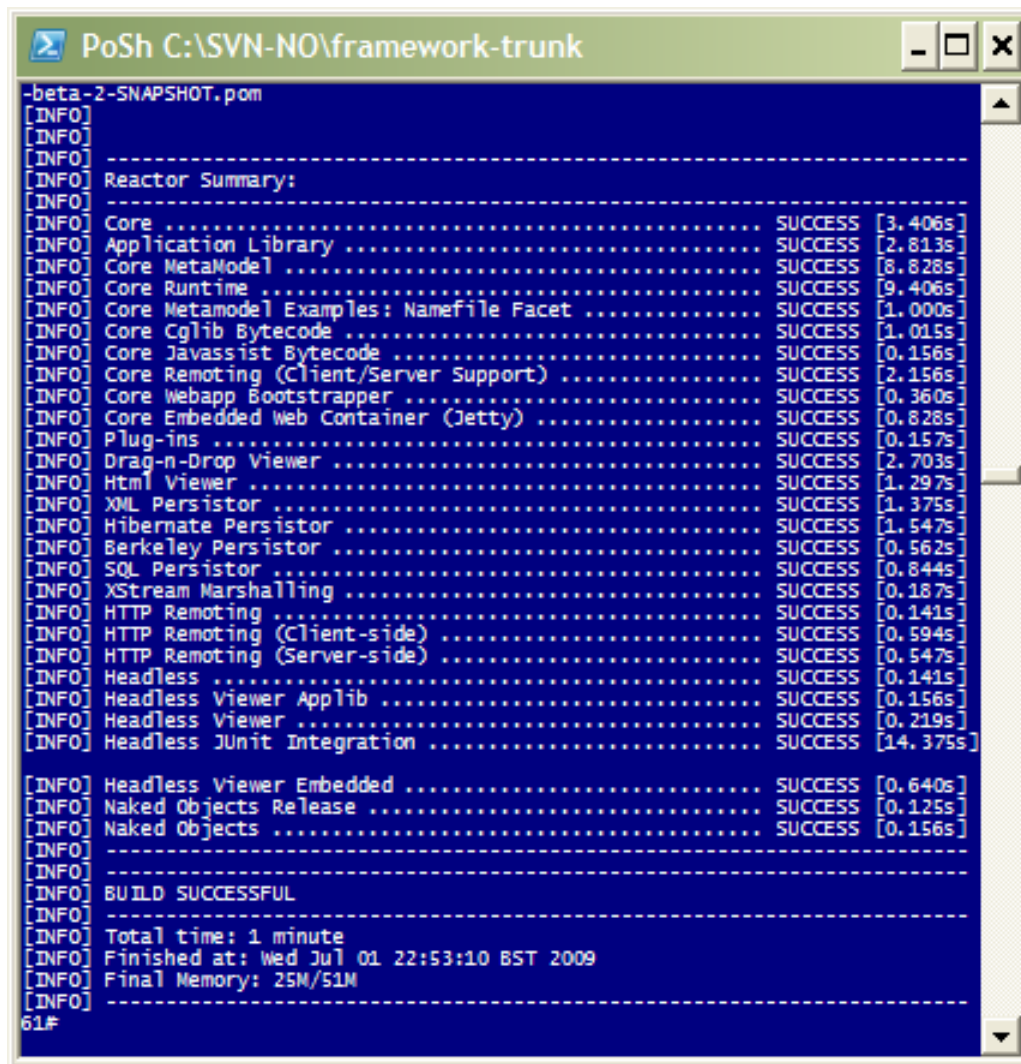
How to build Naked Objects from Source

Building just the Maven-Released Modules (the core framework + plugins)

To build just the Maven released plugins (which, actually, is what you most often need to do):

```
$ cd ../framework/trunk
$ mvn clean install
```

You should end up with:



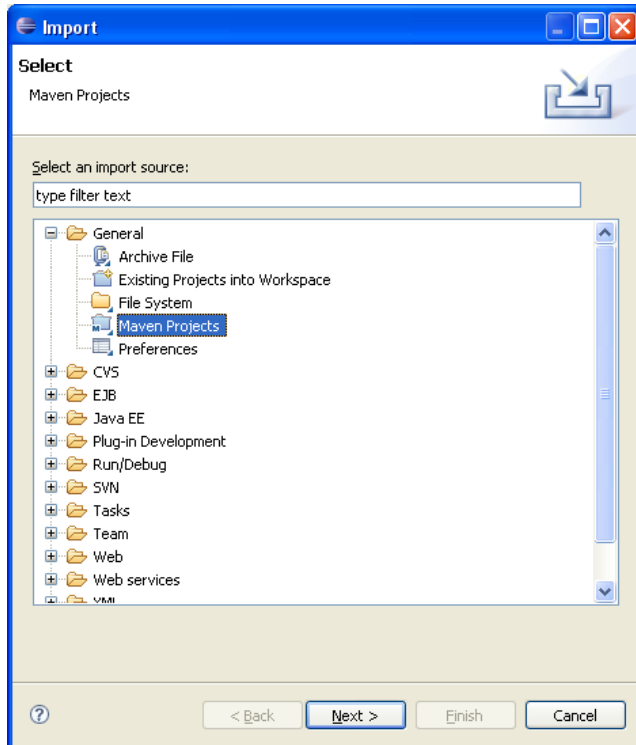
```
-beta-2-SNAPSHOT.pom
[INFO]
[INFO]
[INFO] -----
[INFO] Reactor Summary:
[INFO] -----
[INFO] Core ..... SUCCESS [3.406s]
[INFO] Application Library ..... SUCCESS [2.813s]
[INFO] Core MetaModel ..... SUCCESS [8.828s]
[INFO] Core Runtime ..... SUCCESS [9.406s]
[INFO] Core Metamodel Examples: Namefile Facet ..... SUCCESS [1.000s]
[INFO] Core Cglib Bytecode ..... SUCCESS [1.015s]
[INFO] Core Javassist Bytecode ..... SUCCESS [0.156s]
[INFO] Core Remoting (Client/Server Support) ..... SUCCESS [2.156s]
[INFO] Core Webapp Bootstrapper ..... SUCCESS [0.360s]
[INFO] Core Embedded Web Container (Jetty) ..... SUCCESS [0.828s]
[INFO] Plug-ins ..... SUCCESS [0.157s]
[INFO] Drag-n-Drop Viewer ..... SUCCESS [2.703s]
[INFO] Html Viewer ..... SUCCESS [1.297s]
[INFO] XML Persistor ..... SUCCESS [1.375s]
[INFO] Hibernate Persistor ..... SUCCESS [1.547s]
[INFO] Berkeley Persistor ..... SUCCESS [0.562s]
[INFO] SQL Persistor ..... SUCCESS [0.844s]
[INFO] XStream Marshalling ..... SUCCESS [0.187s]
[INFO] HTTP Remoting ..... SUCCESS [0.141s]
[INFO] HTTP Remoting (Client-side) ..... SUCCESS [0.594s]
[INFO] HTTP Remoting (Server-side) ..... SUCCESS [0.547s]
[INFO] Headless ..... SUCCESS [0.141s]
[INFO] Headless Viewer Applib ..... SUCCESS [0.156s]
[INFO] Headless Viewer ..... SUCCESS [0.219s]
[INFO] Headless JUnit Integration ..... SUCCESS [14.375s]
[INFO] Headless Viewer Embedded ..... SUCCESS [0.640s]
[INFO] Naked Objects Release ..... SUCCESS [0.125s]
[INFO] Naked Objects ..... SUCCESS [0.156s]
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 1 minute
[INFO] Finished at: Wed Jul 01 22:53:10 BST 2009
[INFO] Final Memory: 25M/51M
[INFO] -----
61#
```

How to build Naked Objects from Source

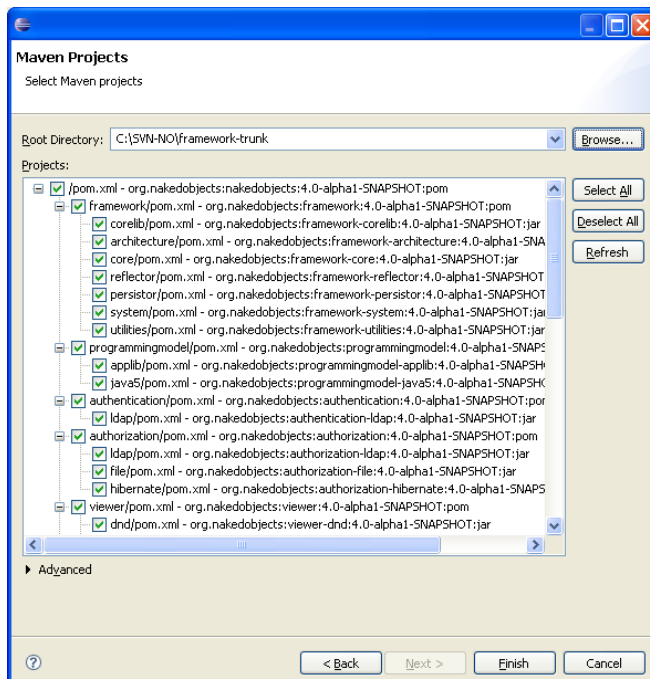
Building in Eclipse

[NB: THESE SCREENSHOTS ARE A LITTLE OUT OF DATE, BUT YOU GET THE IDEA...]

Use File > Import to import the Maven poms.

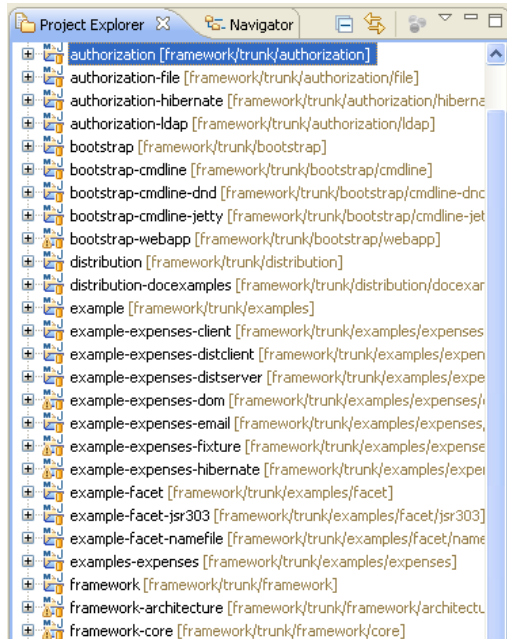


Specify the framework/trunk as root directory. The M2Eclipse plugin will locate all the Maven projects referenced



How to build Naked Objects from Source

Hit finish. You'll end up with a bunch of projects:



You might then want to organize into working sets.

You're now set to develop using either Maven or Eclipse.

Hints and Tips

- switch off automatic builds in Eclipse and build manually
- Using Windows > Preferences > Maven
 - enable offline mode in Eclipse to save checking remote repositories
 - point to your external Maven installation to minimize differences
- whenever you need to build from the commandline as well, use `mvn install -o` rather than `mvn clean install`
 - omitting clean means that Eclipse doesn't lose a handle on the JARs that it is referencing
 - the `-o` flag means run offline, to save checking remote repositories
- if you do a full `mvn clean install`, then do a refresh in Eclipse afterwards
- if M2Eclipse gets its confused, do a full refresh, followed by update dependencies, followed by update project configuration. That usually does the trick.