# Agile Software Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

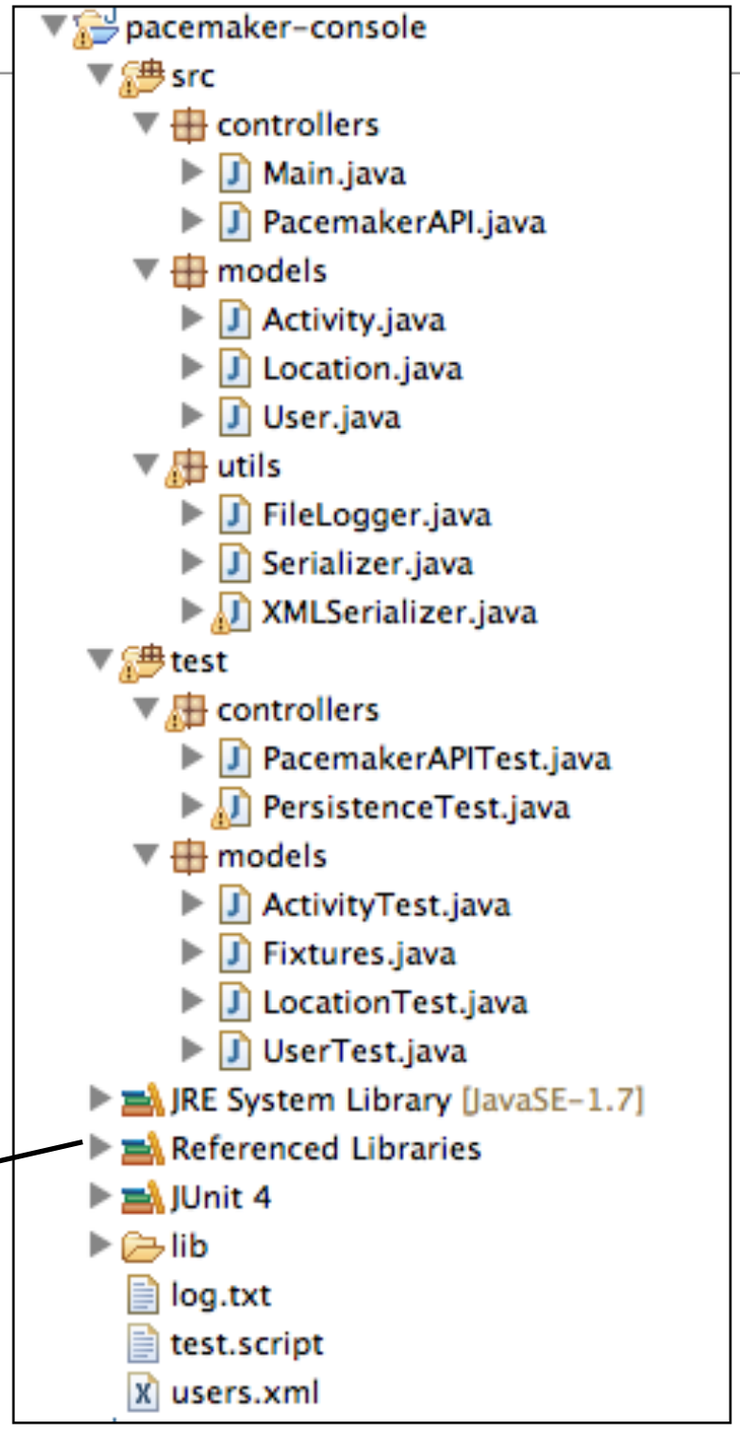
http://elearning.wit.ie

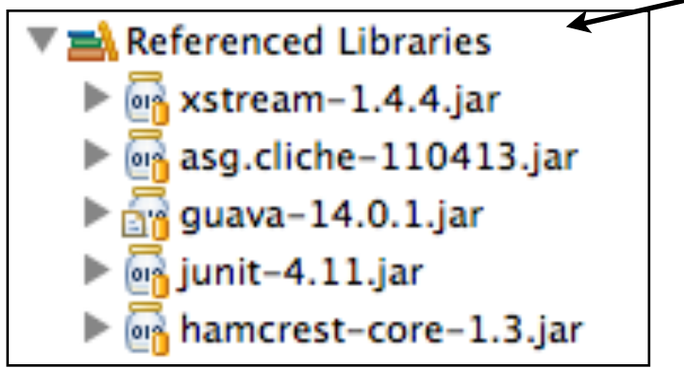
Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

# pacemaker-console

- Take the pacemaker-console (java) project
- Note the external dependencies:
  - guava-14.0.1.jar
  - xstream-1.4.4.jar
  - junit-4.11.jar
  - asg.cliche-11-413.jar



# Project Creation Archetype

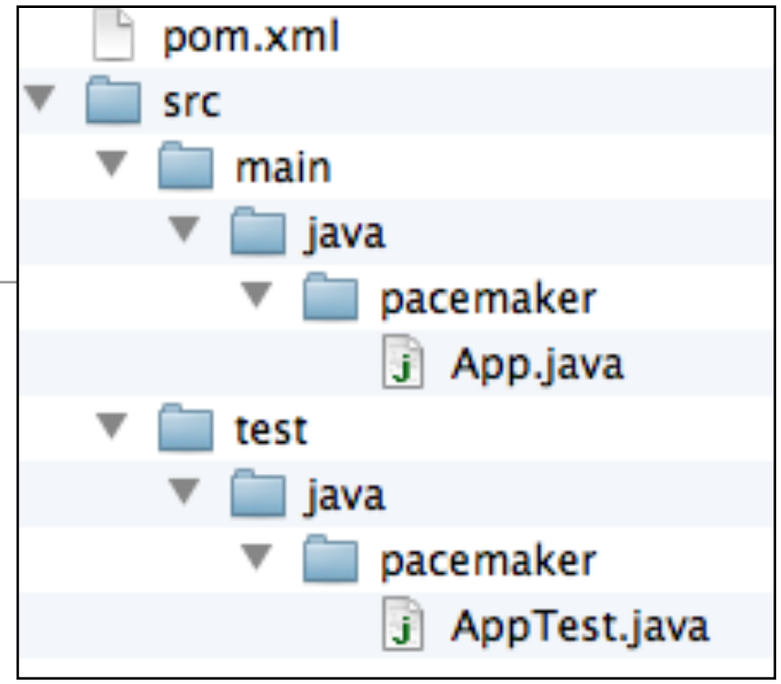
- From the command line, execute maven command to create a project skeleton structure:

```
mvn archetype:create -DarchetypeGroupId=org.apache.maven.archetypes
                     -DarchetypeArtifactId=maven-archetype-quickstart
                     -DgroupId=pacemaker
                     -DartifactId=pacemaker-console-maven
```

```
Grendel:Session 9 edeleastar$ mvn archetype:create -DarchetypeGroupId=org.apache.maven.archetypes -D
archetypeArtifactId=maven-archetype-quickstart -DgroupId=msccomm -DartifactId=Pim
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'archetype'.
[INFO] ------------------------------------------------------------------------
[INFO] Building Maven Default Project
[INFO]    task-segment: [archetype:create] (aggregator-style)
[INFO] ------------------------------------------------------------------------
[INFO] Setting property: classpath.resource.loader.class => 'org.codehaus.plexus.velocity.ContextCla
ssLoaderResourceLoader'.
[INFO] Setting property: velocimacro.messages.on => 'false'.
[INFO] Setting property: resource.loader => 'classpath'.
[INFO] Setting property: resource.manager.logwhenfound => 'false'.
[INFO] *************************************************************
[INFO] Starting Jakarta Velocity v1.4
[INFO] RuntimeInstance initializing.
[INFO] Default Properties File: org/apache/velocity/runtime/defaults/velocity.properties
[INFO] Default ResourceManager initializing. (class org.apache.velocity.runtime.resource.ResourceMan
agerImpl)
[INFO] Resource Loader Instantiated: org.codehaus.plexus.velocity.ContextClassLoaderResourceLoader
[INFO] ClasspathResourceLoader : initialization starting.
[INFO] ClasspathResourceLoader : initialization complete.
[INFO] ResourceCache : initialized. (class org.apache.velocity.runtime.resource.ResourceCacheImpl)
[INFO] Default ResourceManager initialization complete.
[INFO] Loaded System Directive: org.apache.velocity.runtime.directive.Literal
[INFO] Loaded System Directive: org.apache.velocity.runtime.directive.Macro
[INFO] Loaded System Directive: org.apache.velocity.runtime.directive.Parse
[INFO] Loaded System Directive: org.apache.velocity.runtime.directive.Include
[INFO] Loaded System Directive: org.apache.velocity.runtime.directive.Foreach
[INFO] Created: 20 parsers.
[INFO] Velocimacro : initialization starting.
[INFO] Velocimacro : adding VMs from VM library template : VM_global_library.vm
[ERROR] ResourceManager : unable to find resource 'VM_global_library.vm' in any resource loader.
[INFO] Velocimacro : error using  VM library template VM_global_library.vm : org.apache.velocity.exc
eption.ResourceNotFoundException: Unable to find resource 'VM_global_library.vm'
[INFO] Velocimacro :  VM library template macro registration complete.
                                                                                templates
                                                                 may NOT replace previous VM

                                                                be  global in scope if allow
                                                                                ------------
[INFO] Using following parameters for creating Archetype: maven-archetype-quickstart:RELEASE
[INFO] ------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: msccomm
[INFO] Parameter: packageName, Value: msccomm
[INFO] Parameter: basedir, Value: /Users/edeleastar/svn/CourseWork/2008/Agile Software Development/s
rc/Session 9
[INFO] Parameter: package, Value: msccomm
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: artifactId, Value: Pim
[INFO] ******************** End of debug info from resources from generated POM ******************
****
[INFO] Archetype created in dir: /Users/edeleastar/svn/CourseWork/2008/Agile Software Development/sr
c/Session 9/Pim
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESSFUL
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2 seconds
[INFO] Finished at: Sat Nov 08 10:46:09 GMT 2008
[INFO] Final Memory: 5M/9M
[INFO] ------------------------------------------------------------------------
```

# Generated Directory Structure

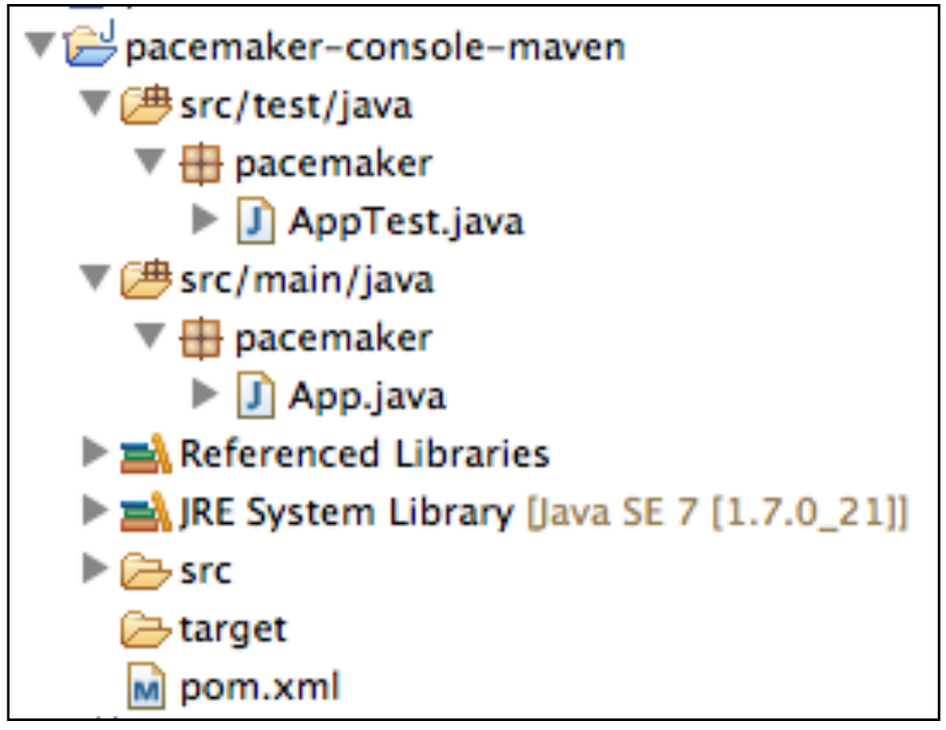
- The generated directory structure contains two dummy java files that can be discarded
- It also generates a POM, which is the basis for the project dependency structure.




```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

# Generate ‘Eclipse’ project

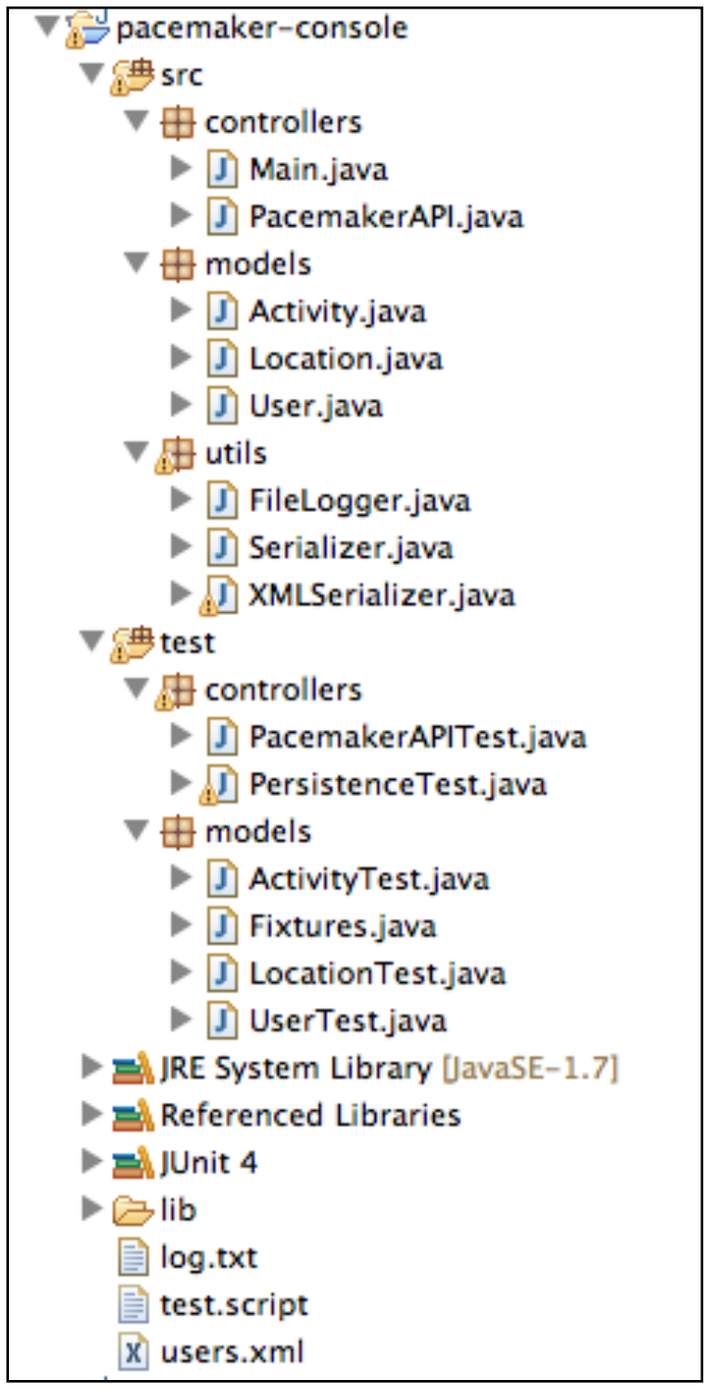
```
mvn eclipse:eclipse
```



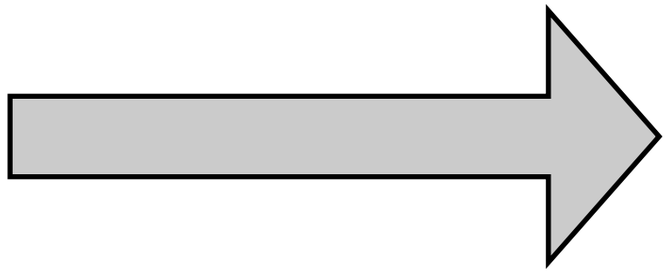

- Generates the ‘.project’ and ‘.classpath’ Eclipse uses to specify project name, structure and dependencies.

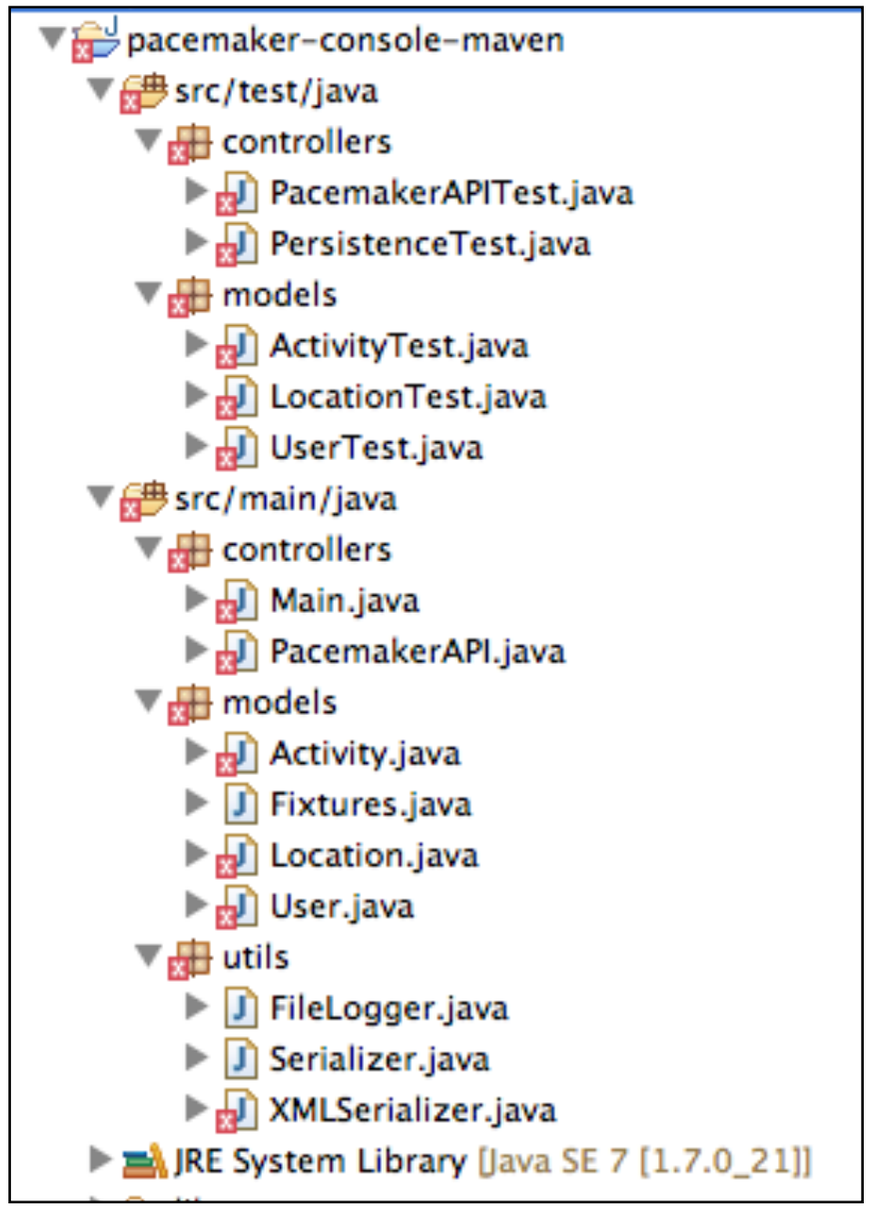
- The project can then be ‘imported’ into eclipse

# Manipulate the project in eclipse...

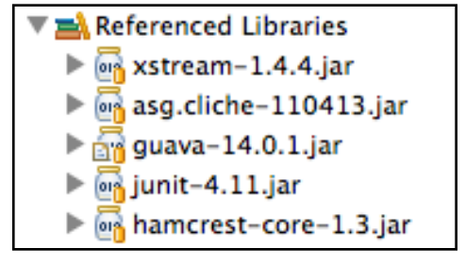



- Copy Paste sources from original project into maven generated project.






Reference errors - project is missing ‘Referenced Libraries’



# Default POM

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

# url + version

```
  <version>1.0</version>
  <url>http://www.wit.ie</url>
```

```
<project...>
  <modelVersion>4.0.0</modelVersion>

  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://www.wit.ie</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

# Java 7 Support + ‘rev’ Junit to 4.11

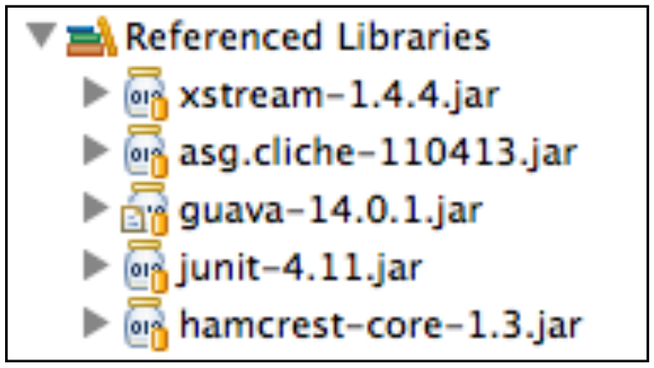
```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.7</maven.compiler.source>
  <maven.compiler.target>1.7</maven.compiler.target>
</properties>
```

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.11</version>
  <scope>test</scope>
</dependency>
```

```xml
<project ...>

  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://www.wit.ie</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>

    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

  </dependencies>

</project>
```

# Dependencies

- Junit installed
- Need to incorporate
  - guava
  - xstream
- and also
  - asg.cliche
- what about
  - hamcrest-core?

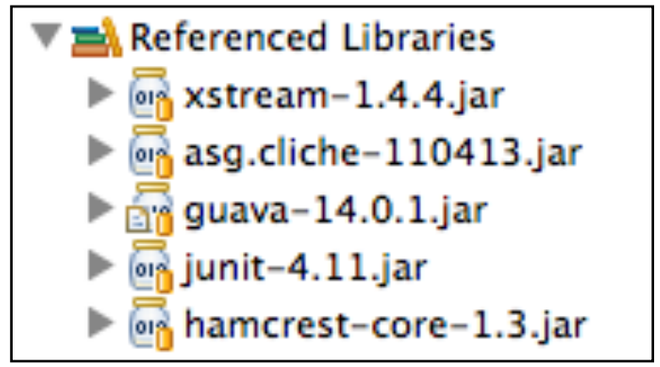



- Well supported, maintained components:
  - xstream, guava
- abandoned, legacy?
  - asg.cliche
- Downstream dependency of junit
  - hamcrest-core

# Dependency entries:

```
<dependencies>

  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <version>15.0</version>
  </dependency>

  <dependency>
    <groupId>com.thoughtworks.xstream</groupId>
    <artifactId>xstream</artifactId>
    <version>1.4.4</version>
  </dependency>

  <dependency>
    <groupId>asg-cliche</groupId>
    <artifactId>asg-cliche</artifactId>
    <version>1.0</version>
  </dependency>

</dependencies>
```

- Junit
- guava
- xstream
- asg.cliche




junit ‘depends on’ hamcrest’, and including junit will automatically incorporate hamcrest into build

```xml
<project ...>
  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://www.wit.ie</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>com.google.guava</groupId>
      <artifactId>guava</artifactId>
      <version>15.0</version>
    </dependency>

    <dependency>
      <groupId>com.thoughtworks.xstream</groupId>
      <artifactId>xstream</artifactId>
      <version>1.4.4</version>
    </dependency>

    <dependency>
      <groupId>asg-cliche</groupId>
      <artifactId>asg-cliche</artifactId>
      <version>1.0</version>
    </dependency>
  </dependencies>
</project>
```

# Complete POM

- Project identifiers
- Project version
- Language version
- Dependencies

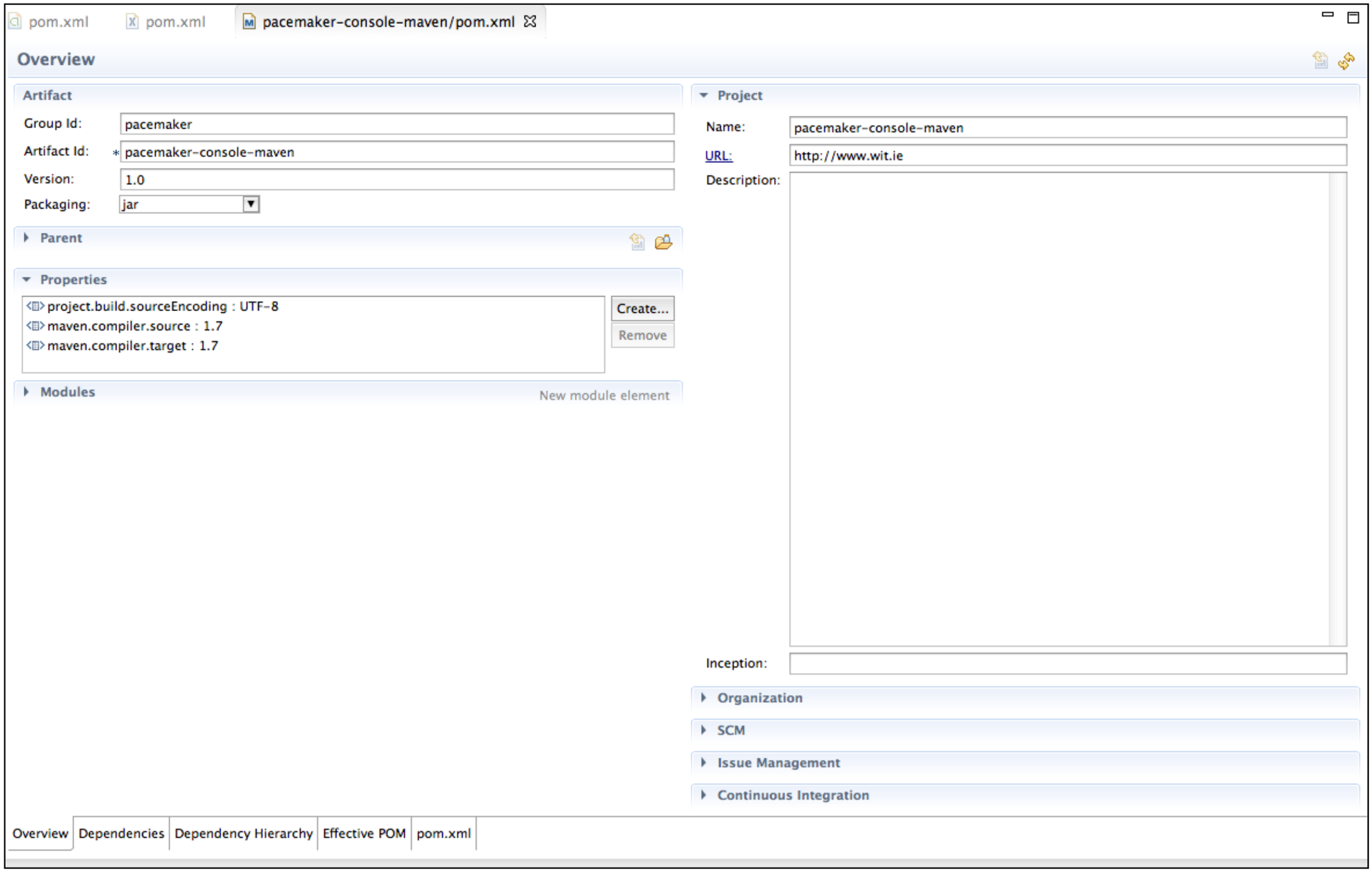
# Design View (Eclipse XML Editor)

pom.xml | pom.xml

| Node | Content |
|---|---|
| project | all(parent?, modelVersion?, groupId?, artifactId?, packaging?, name?, version?, description?, url?, prerequisites?, issueManageme… |
| xmlns | http://maven.apache.org/POM/4.0.0 |
| xmlns:xsi | http://www.w3.org/2001/XMLSchema-instance |
| xsi:schemaLocation | http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd |
| modelVersion | 4.0.0 |
| groupId | pacemaker |
| artifactId | pacemaker-console-maven |
| version | 1.0 |
| packaging | jar |
| name | pacemaker-console-maven |
| url | http://www.wit.ie |
| properties | (namespace:uri="##any") |
| project.build.sourceEncoding | UTF-8 |
| maven.compiler.source | 1.7 |
| maven.compiler.target | 1.7 |
| dependencies | (dependency*) |
| dependency | all(groupId?, artifactId?, version?, type?, classifier?, scope?, systemPath?, exclusions?, optional?) |
| dependency | all(groupId?, artifactId?, version?, type?, classifier?, scope?, systemPath?, exclusions?, optional?) |
| dependency | all(groupId?, artifactId?, version?, type?, classifier?, scope?, systemPath?, exclusions?, optional?) |
| dependency | all(groupId?, artifactId?, version?, type?, classifier?, scope?, systemPath?, exclusions?, optional?) |

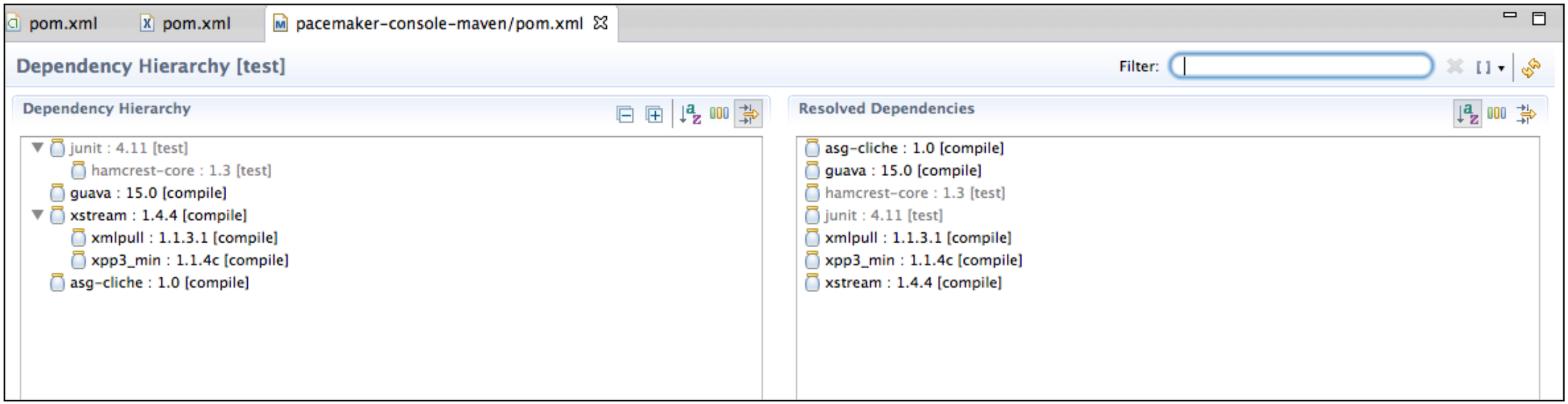
Design | Source

# Eclipse Maven Support (via plugin)



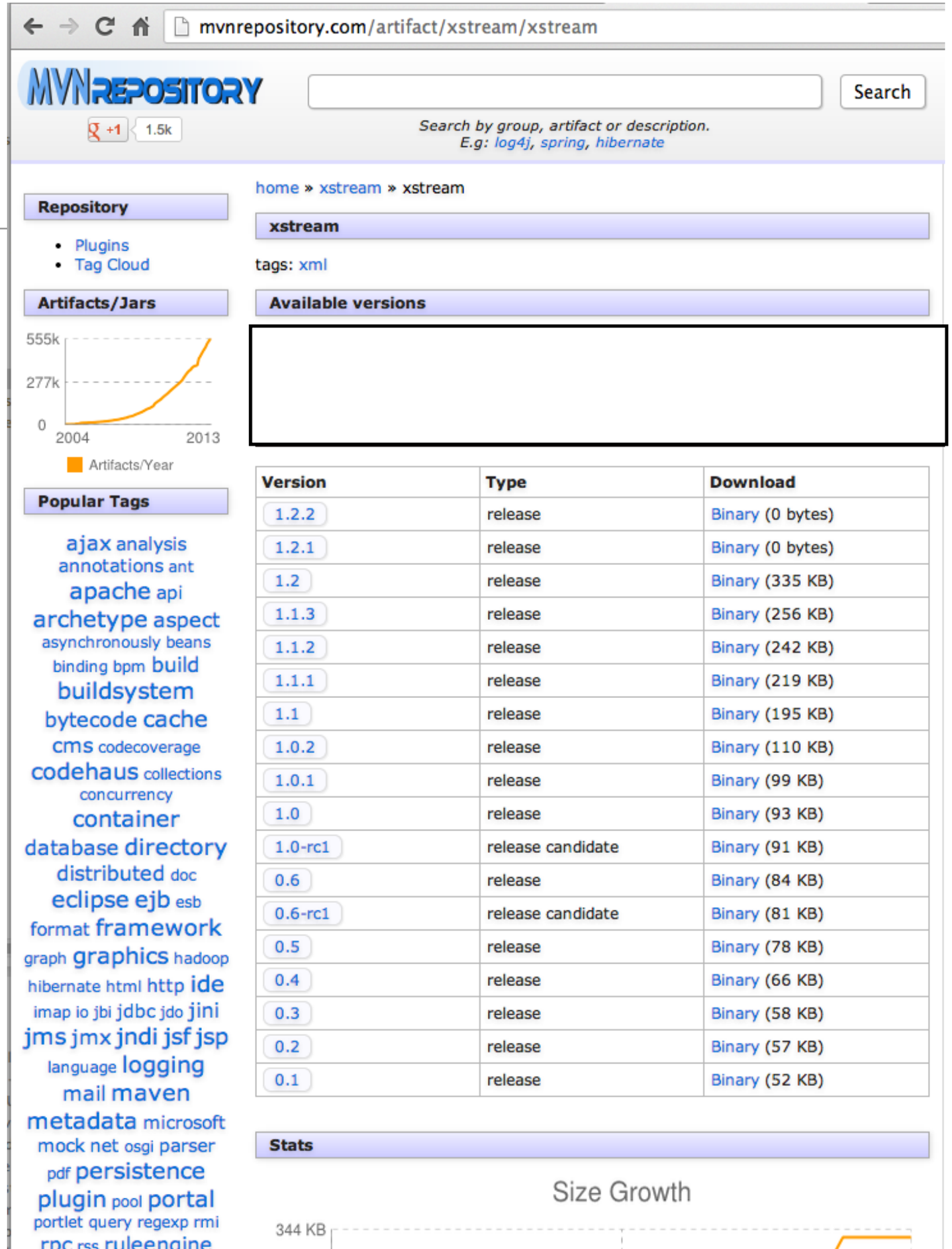
# Dependency Hierarchy

- Shows ‘implicit’ dependencies
  - hamcrest, xmlpull, xpp3_min



# Supported Libraries

- junit, guava and xstream are all under active development, and latest versions are deposited in public maven repositories



# Unsupported Libraries

- asg-cliche is not under active development, and is not in any public maven repo.

- To keep our build consistent, we install asg-cliche in our local repo:

```
mvn install:install-file -Dfile=asg-cliche-1.0.jar -DgroupId=asg-cliche -DartifactId=asg-cliche -Dversion=1.0 -Dpackaging=jar
```

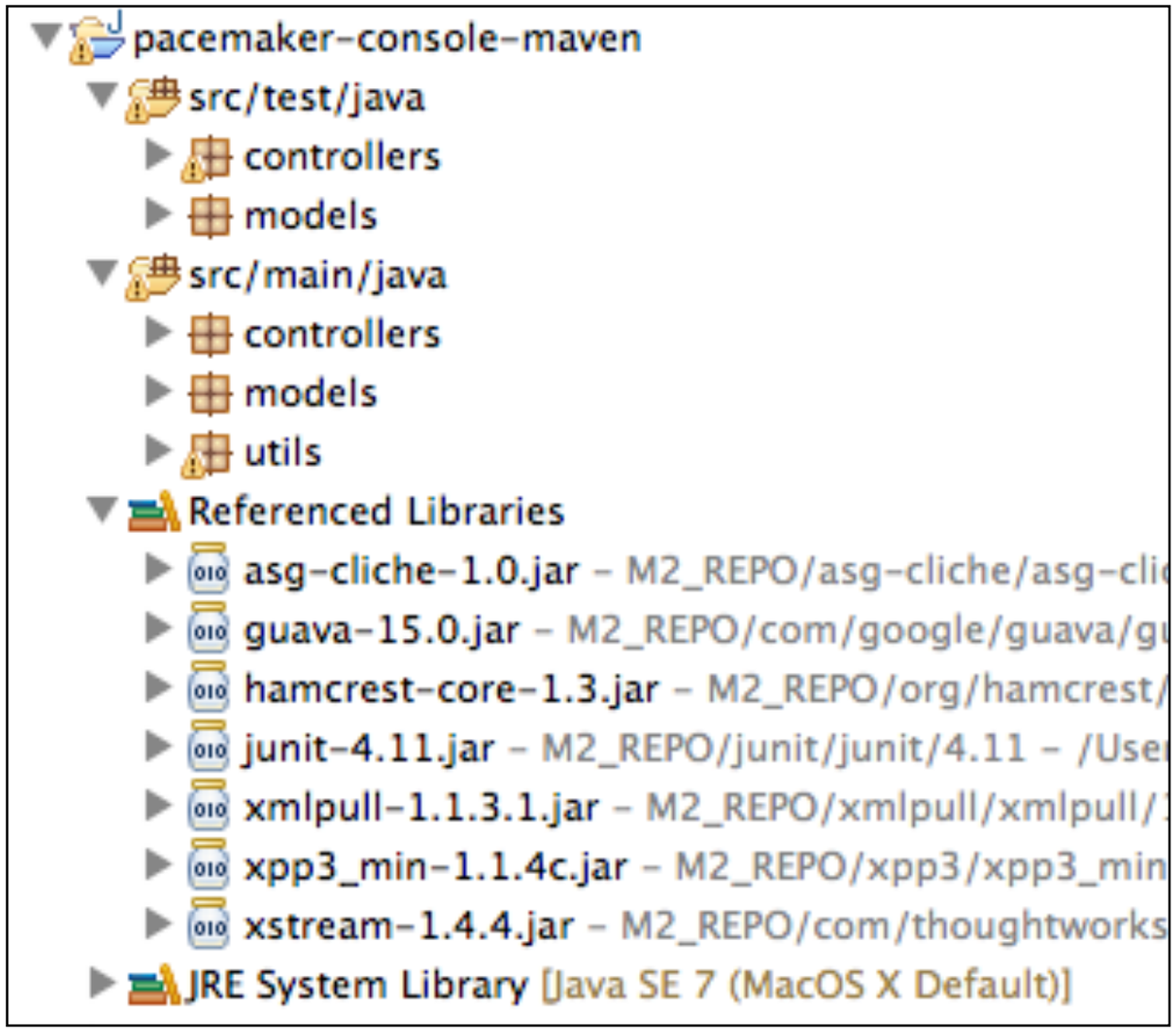
- Our maven build will find it locally, and resolve to that version without searching further.

# Regenerate Eclipse Version

- On command line, instruct maven to refresh the eclipse project

```
mvn eclipse:eclipse
```

This will now include dependancies, including inferred upstream dependencies



# Maven Lifecycles

- Maven is based around the central concept of a build lifecycle - a clearly defined process for building and distributing a particular artifact
- Only necessary to learn a small set of commands to build any Maven project, and the POM will ensure correct execution
- There are three built-in build lifecycles:
  - default: handles your project deployment
  - clean: handles project re-initialization/clean up
  - site. handles the creation of the project's documentation

# Maven Default Lifecycle

- Each of these build lifecycles is defined by a different list of build phases, wherein a build phase represents a stage in the lifecycle.

- For example, the default lifecycle has the following build phases (for a complete list of the build phases, refer to the Lifecycle Reference):
  - validate
  - compile
  - test
  - package
  - integration-test
  - verify
  - install
  - deploy

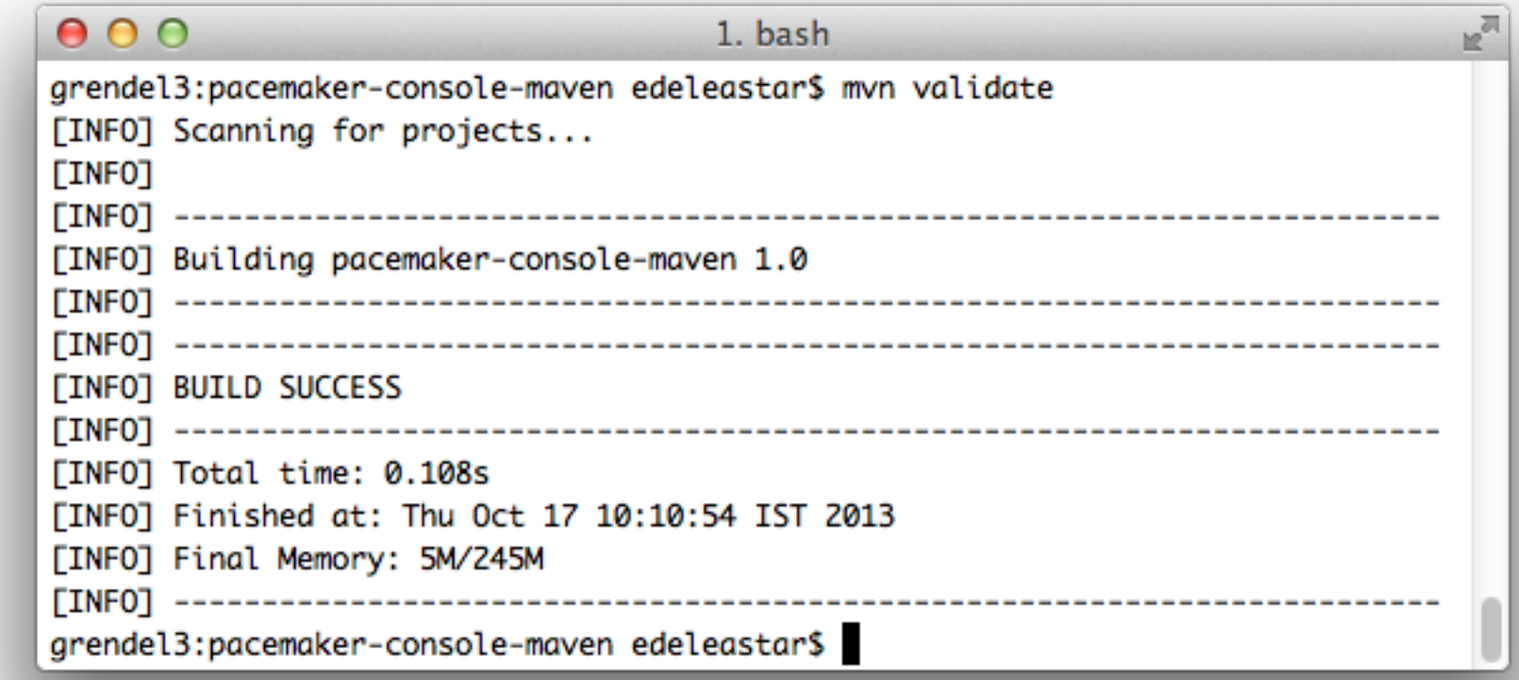
# pacemaker

- Validate
- Compile
- Test
- Package
- Install

```
<project ...>
  <groupId>pacemaker</groupId>
  <artifactId>pacemaker-console-maven</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>

  <name>pacemaker-console-maven</name>
  <url>http://www.wit.ie</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>

    <dependency>
      <groupId>com.google.guava</groupId>
      <artifactId>guava</artifactId>
      <version>15.0</version>
    </dependency>

    <dependency>
      <groupId>com.thoughtworks.xstream</groupId>
      <artifactId>xstream</artifactId>
      <version>1.4.4</version>
    </dependency>

    <dependency>
      <groupId>asg-cliche</groupId>
      <artifactId>asg-cliche</artifactId>
      <version>1.0</version>
    </dependency>
  </dependencies>
</project>
```

# Validate

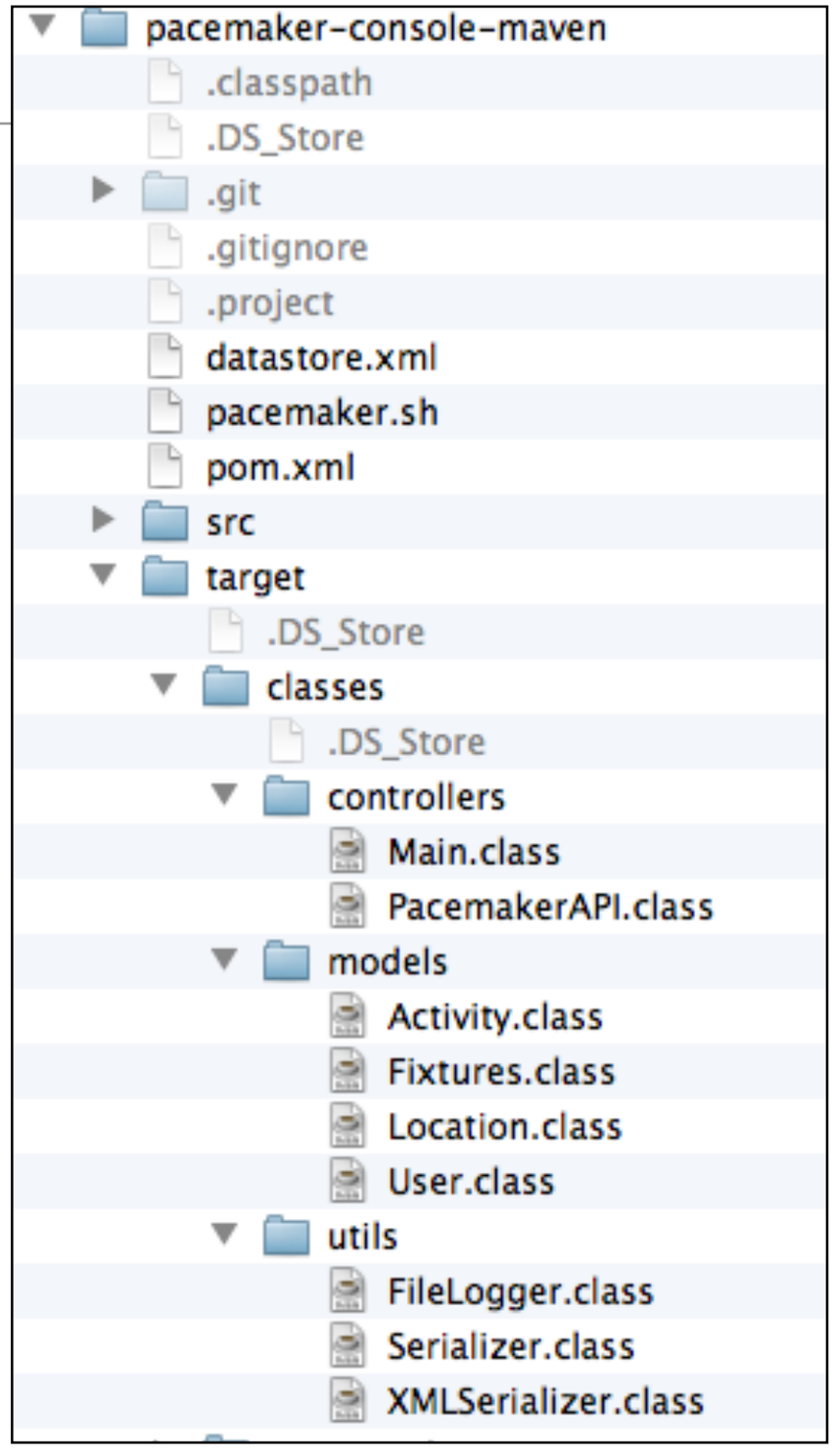
- Validate the project is correct and all necessary information is available



# Compile

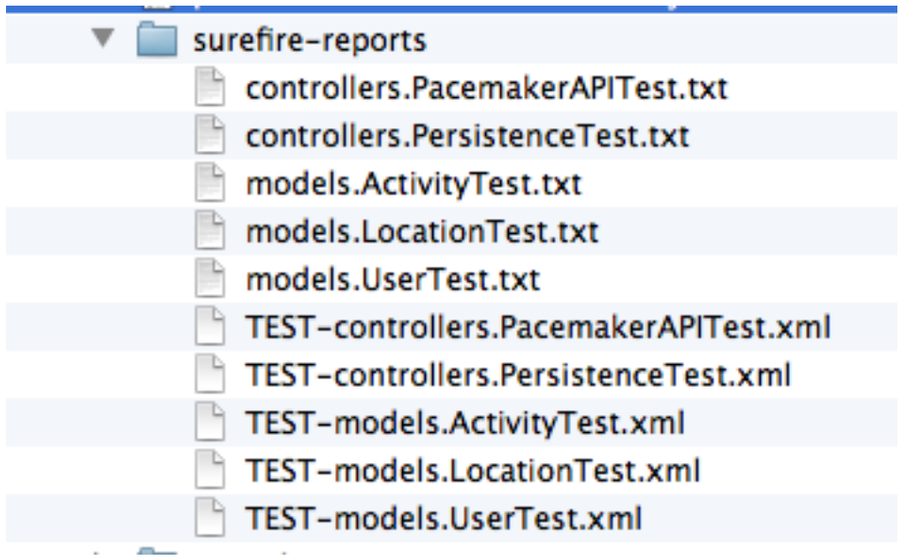
- compile the source code of the project

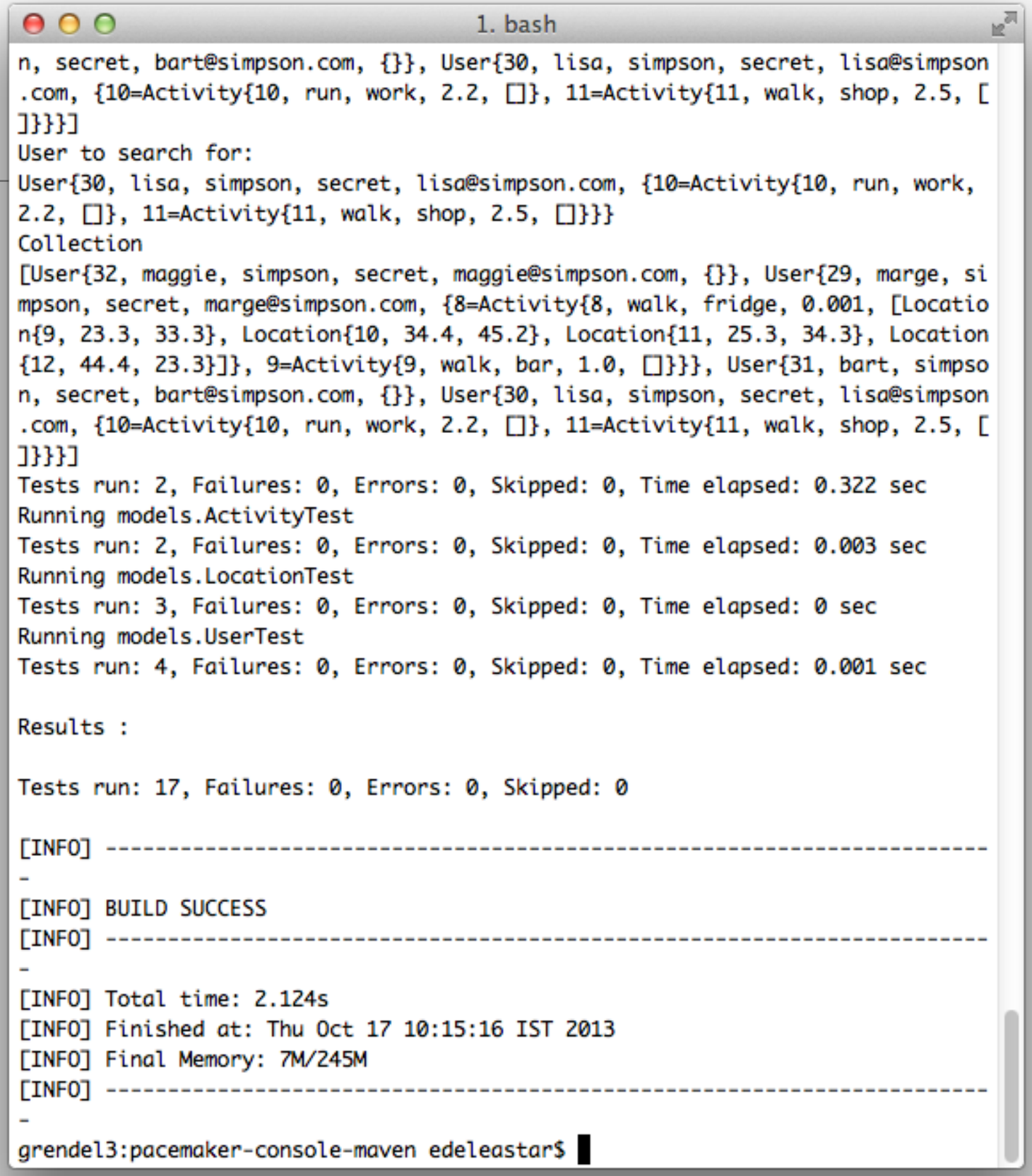
```
1. bash
grendel3:pacemaker-console-maven edeleastar$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building pacemaker-console-maven 1.0
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ pacemaker-
console-maven ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /Users/edeleastar/repos/modules/agile
/prj/pacemaker-console-maven/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ pacemaker-con
sole-maven ---
[INFO] Nothing to compile - all classes are up to date
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 1.737s
[INFO] Finished at: Thu Oct 17 10:12:34 IST 2013
[INFO] Final Memory: 8M/245M
[INFO] ------------------------------------------------------------------------
grendel3:pacemaker-console-maven edeleastar$
```



# Test

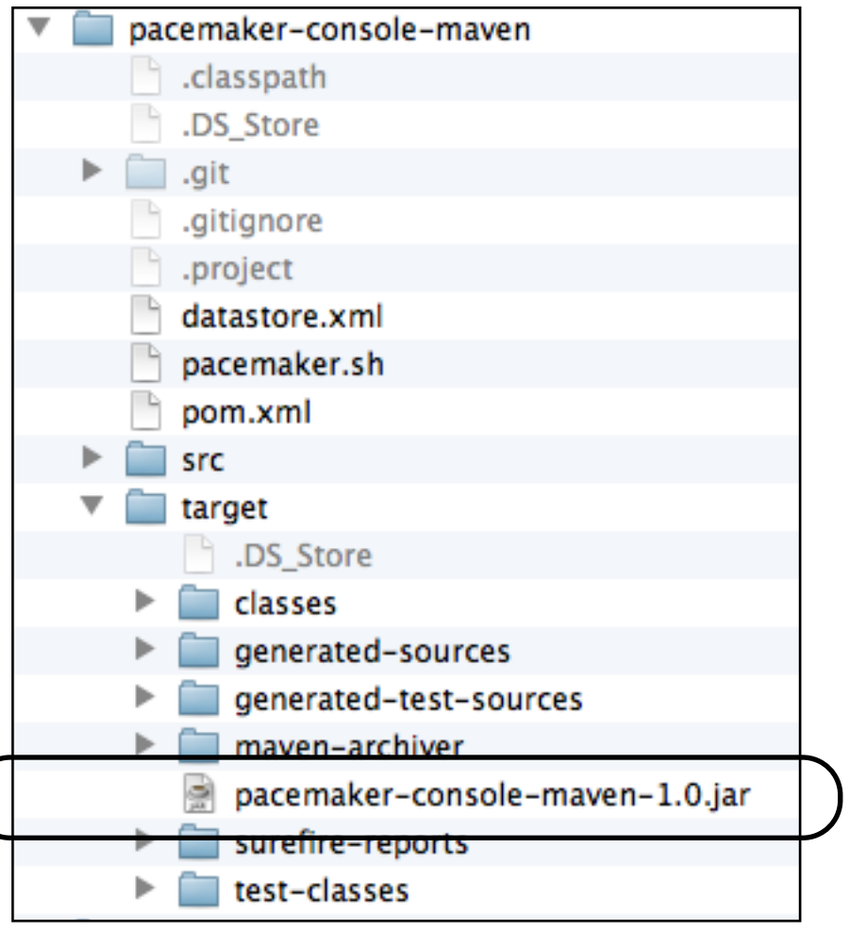
- test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

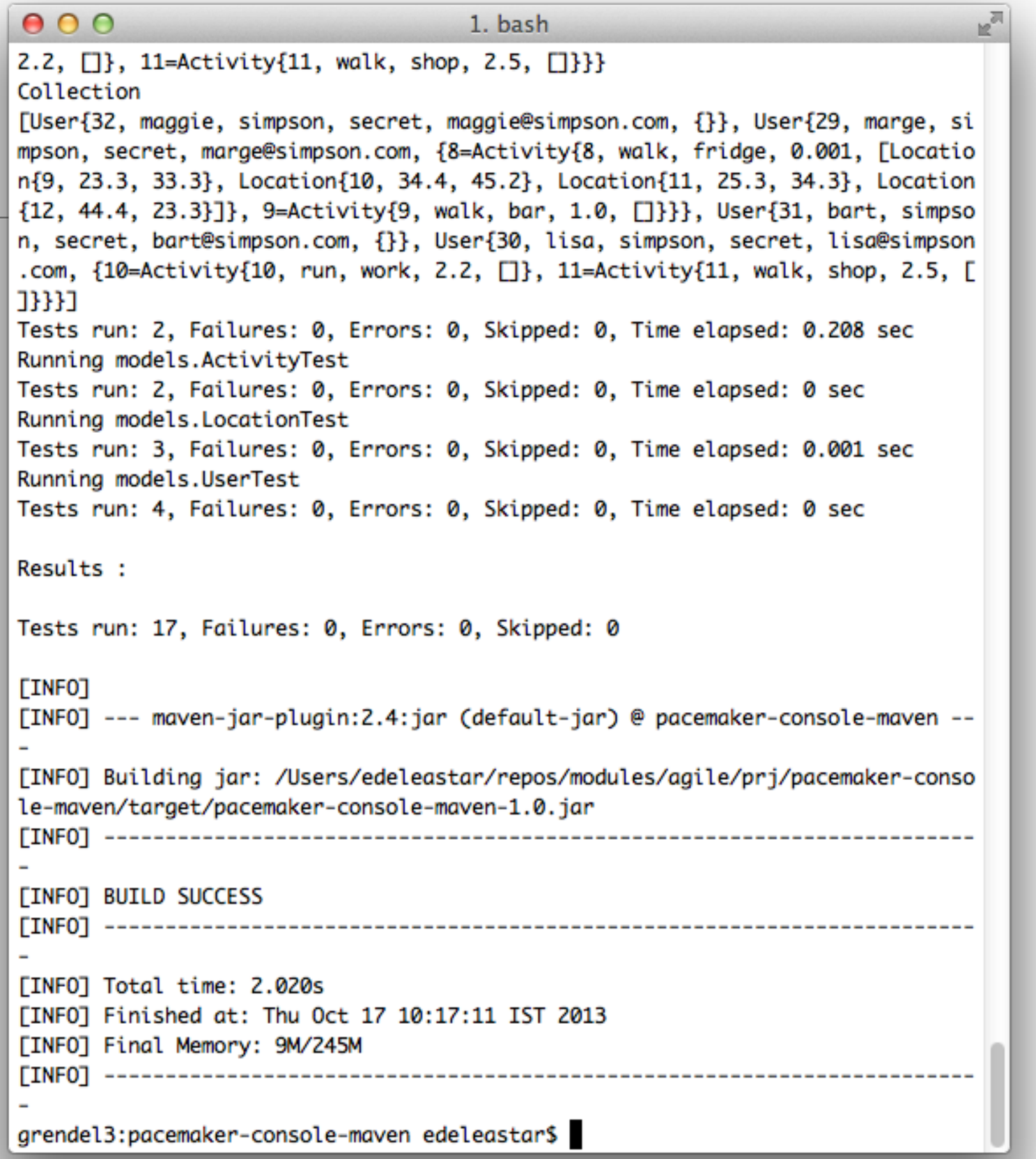


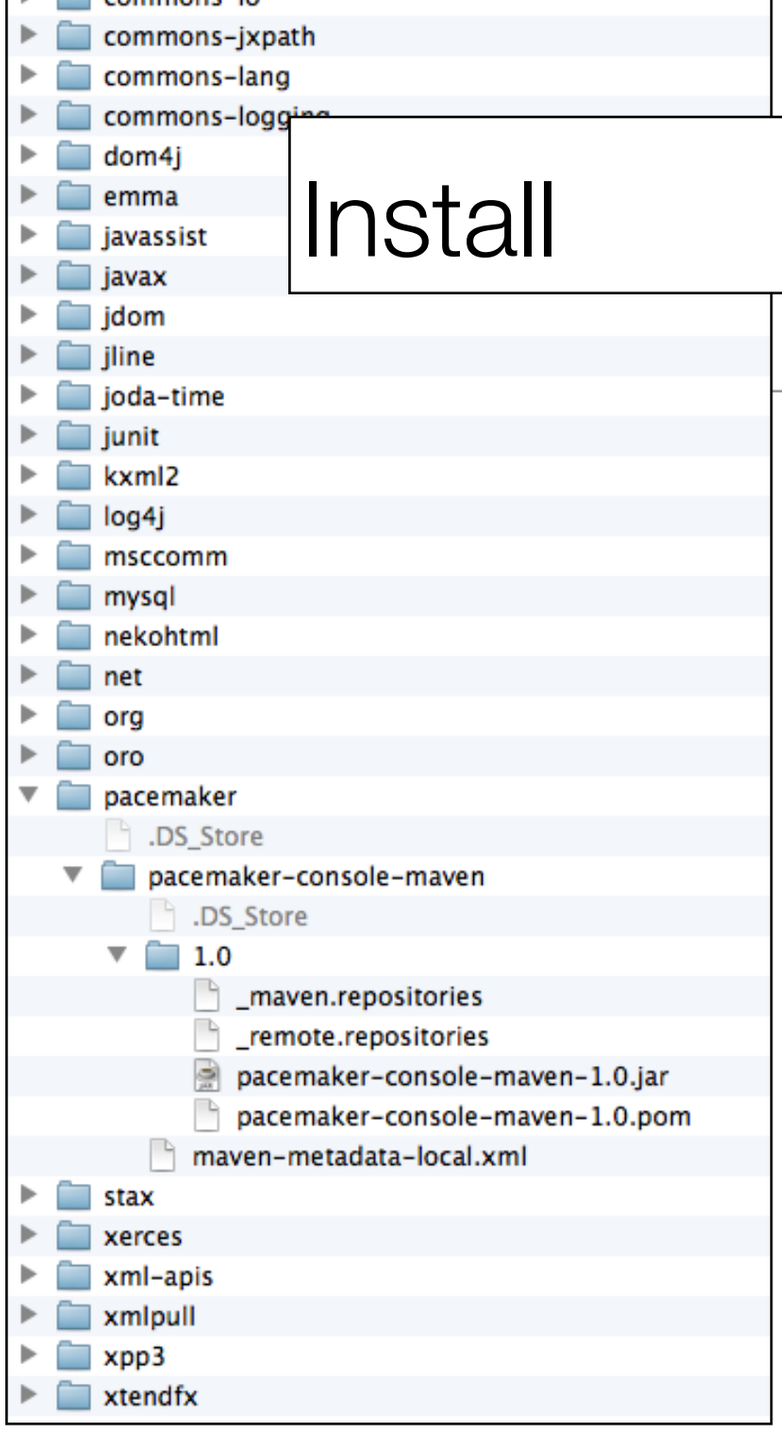
# Package

- take the compiled code and package it in its distributable format, such as a JAR.

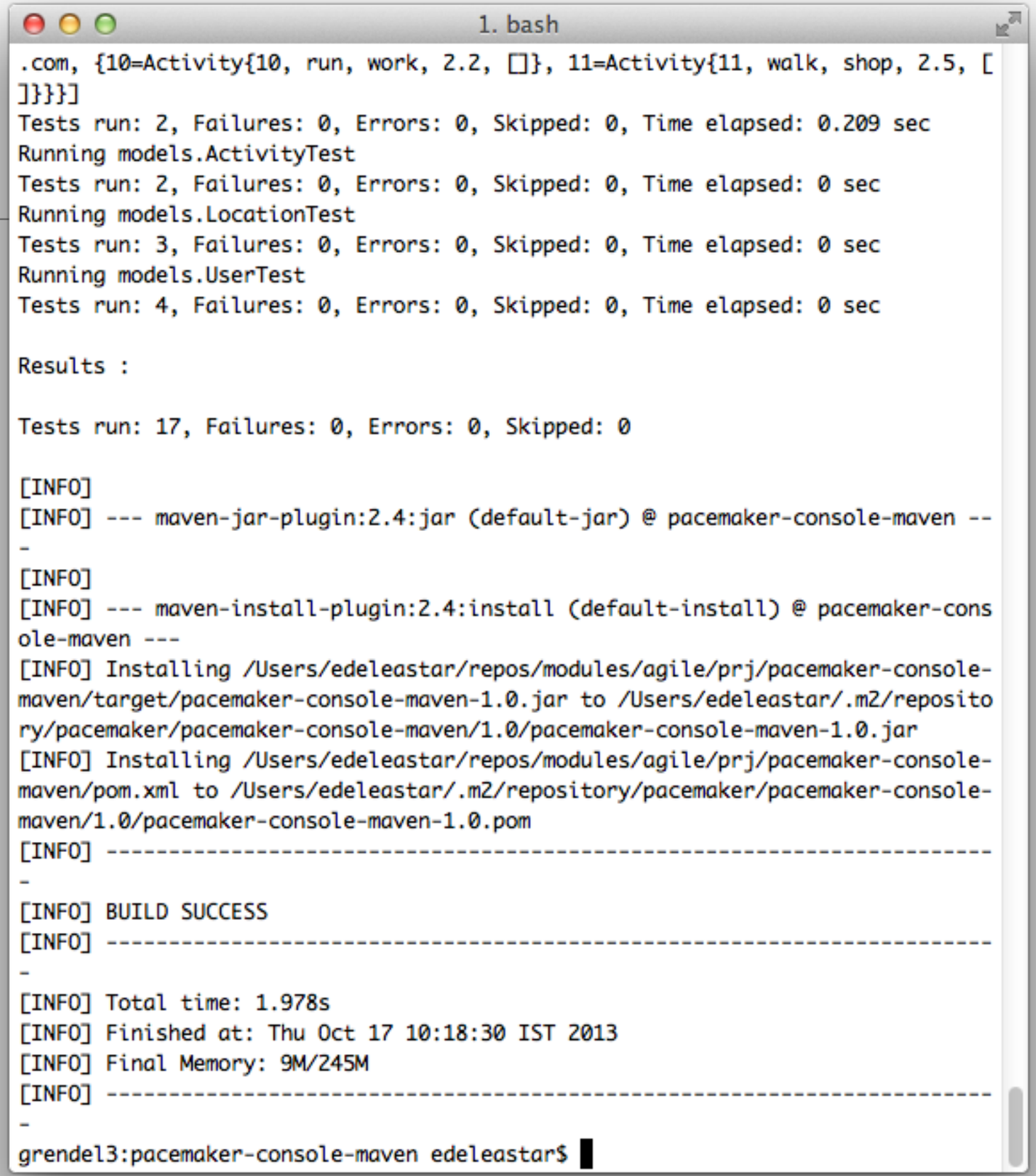


# Install




- install the package into the local repository, for use as a dependency in other projects locally

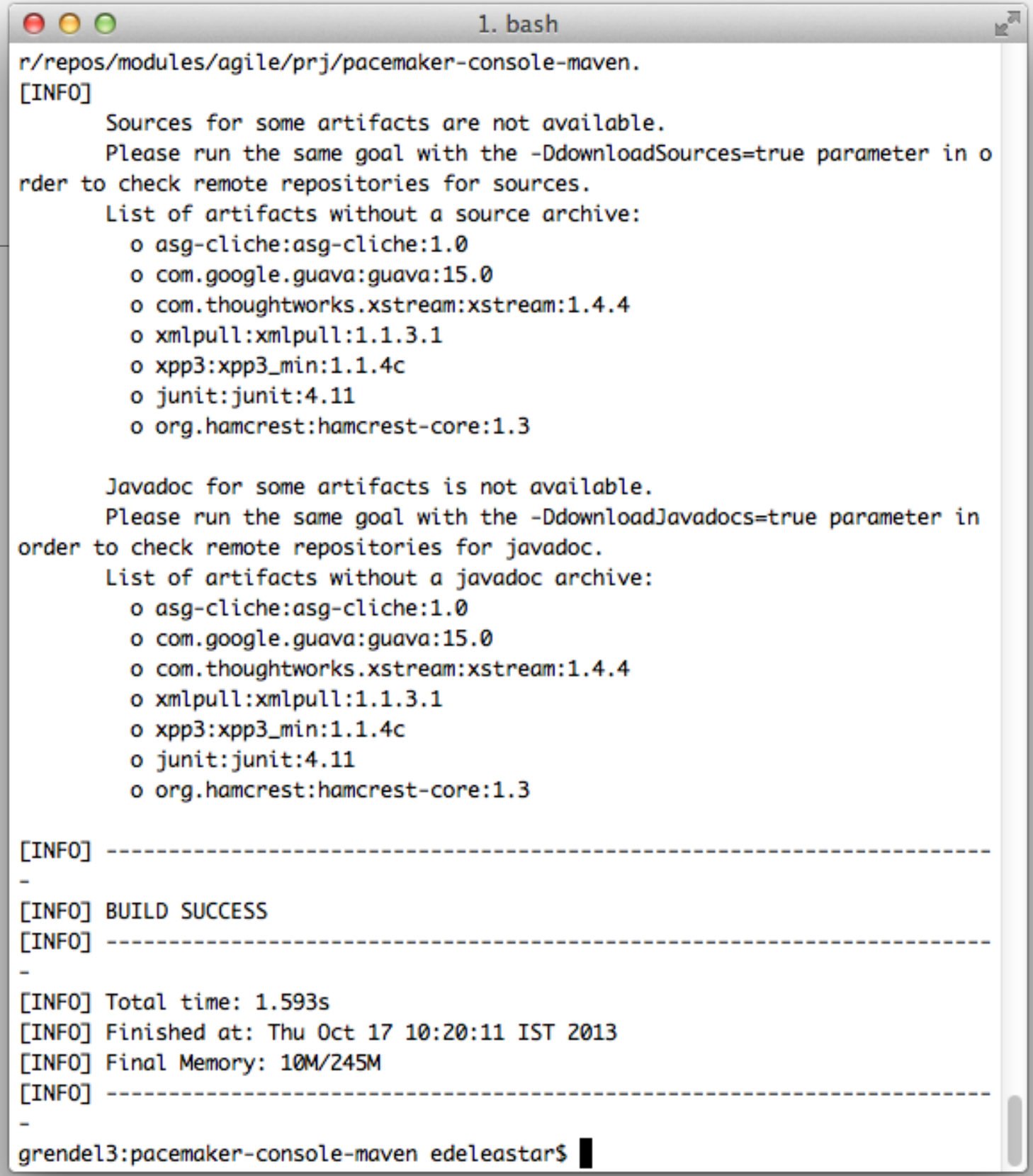


# Maven & Eclipse

- The projects is now built entirely by the maven build system. However, we can continue to use Eclipse for day to day development.

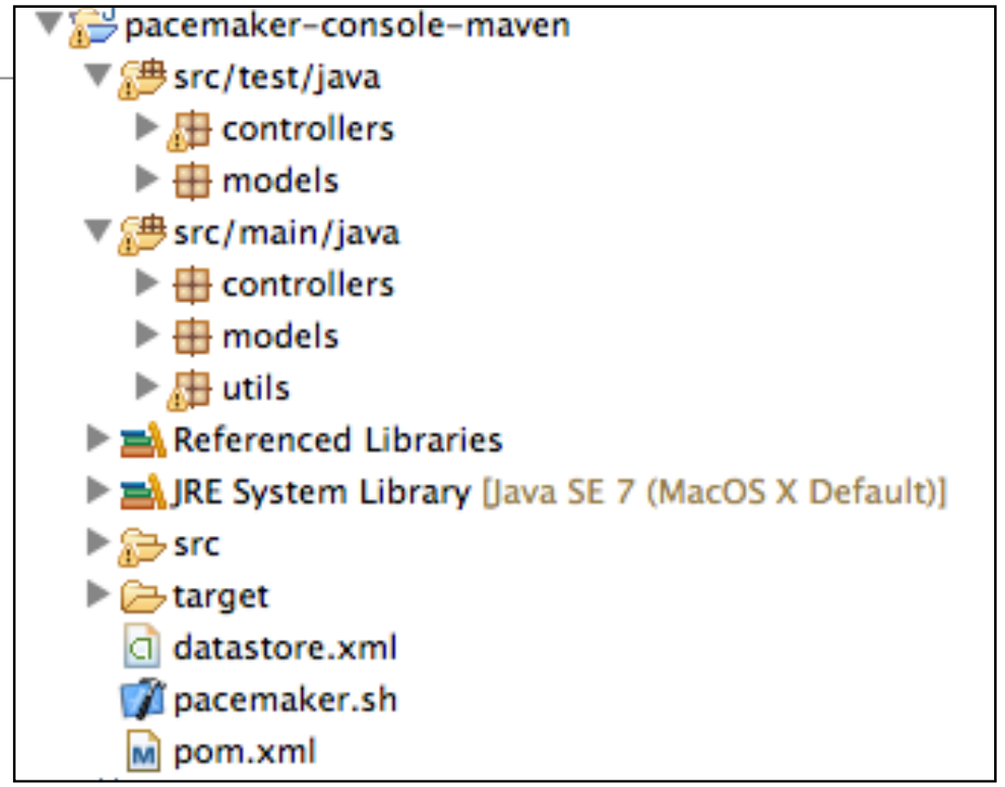
- To do this, we generate an eclipse project from the maven project structure. Do this by issuing the following maven command:

```
mvn eclipse:eclipse
```

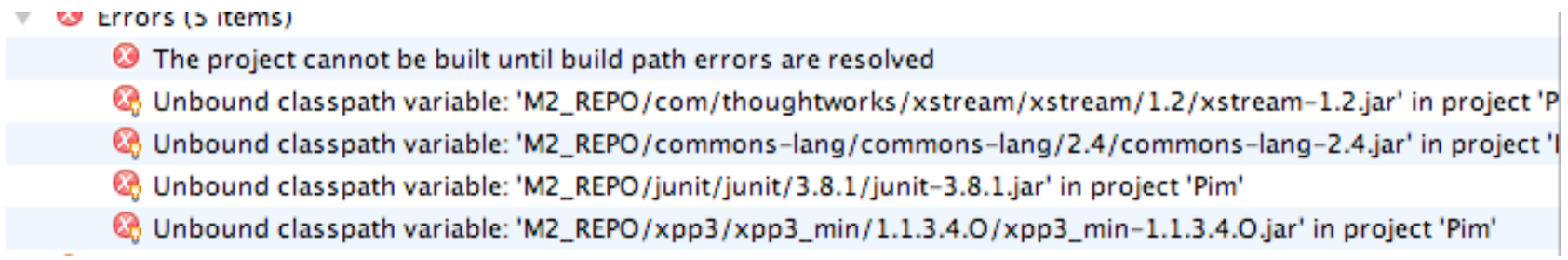


# Eclipse Project Structure

- This will have produced the required .classpath and .project files in the pim directory. You can now import this project into eclipse using the normal import->project menus.




- Although will import successfully, it will not build

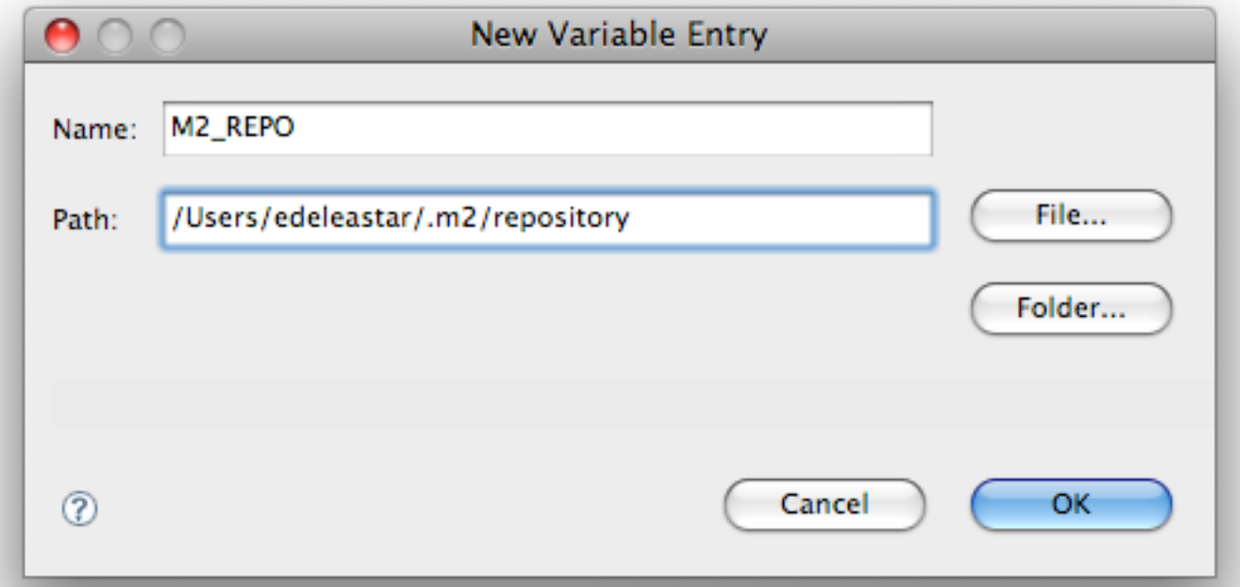


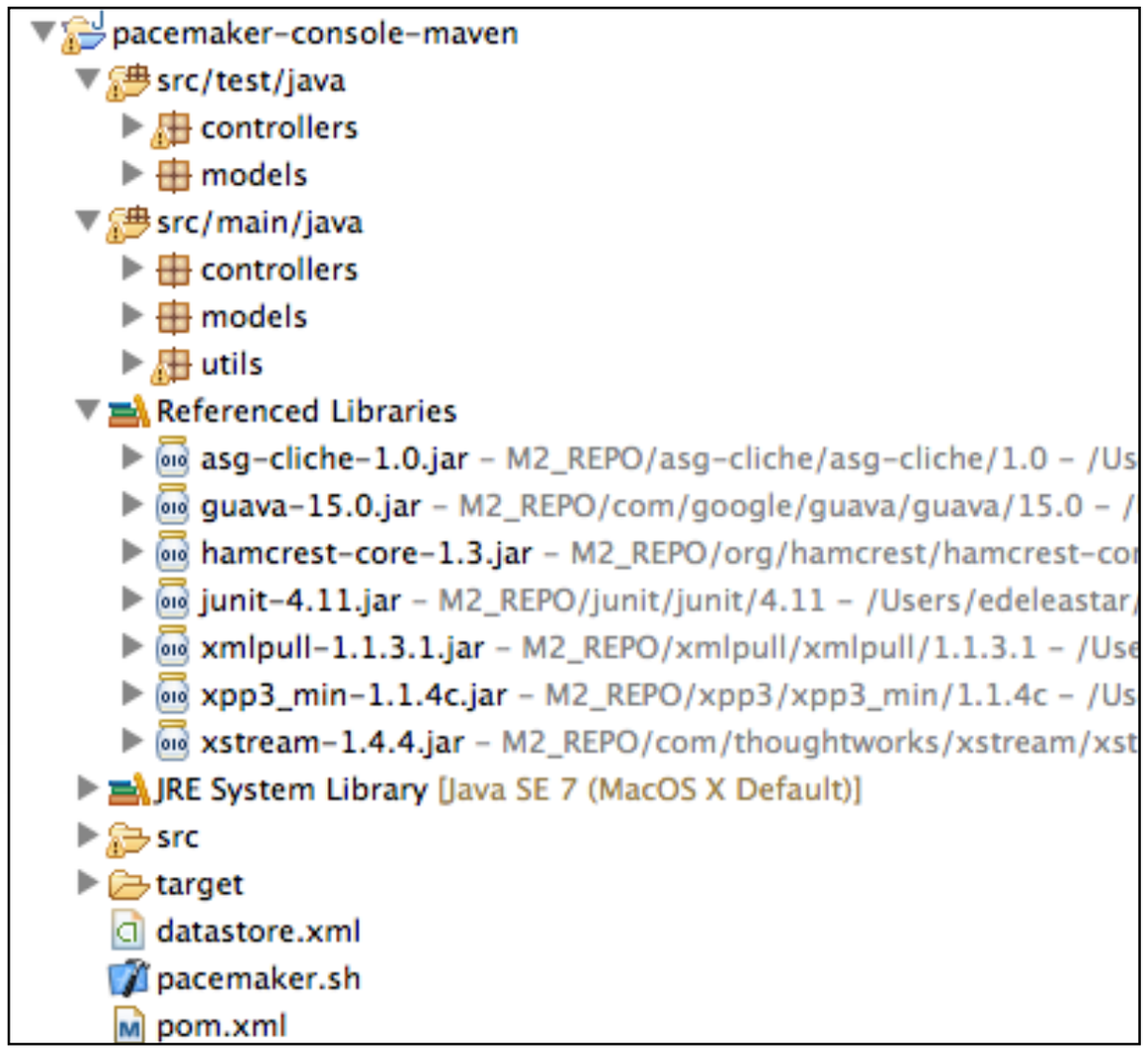
# Eclipse .classpath and .project files

```
<classpath>
  <classpathentry kind="src" path="src/test/java" output="target/test-classes" including="**/*.java"/>
  <classpathentry kind="src" path="src/main/java" including="**/*.java"/>
  <classpathentry kind="output" path="target/classes"/>
  <classpathentry kind="var" path="M2_REPO/asg-cliche/asg-cliche/1.0/asg-cliche-1.0.jar"/>
  <classpathentry kind="var" path="M2_REPO/com/google/guava/guava/15.0/guava-15.0.jar"/>
  <classpathentry kind="var" path="M2_REPO/org/hamcrest/hamcrest-core/1.3/hamcrest-core-1.3.jar"/>
  <classpathentry kind="var" path="M2_REPO/junit/junit/4.11/junit-4.11.jar"/>
  <classpathentry kind="var" path="M2_REPO/xmlpull/xmlpull/1.1.3.1/xmlpull-1.1.3.1.jar"/>
  <classpathentry kind="var" path="M2_REPO/xpp3/xpp3_min/1.1.4c/xpp3_min-1.1.4c.jar"/>
  <classpathentry kind="var" path="M2_REPO/com/thoughtworks/xstream/xstream/1.4.4/xstream-1.4.4.jar"/>
  <classpathentry kind="con" path="org.eclipse.jdt.launching.JRE_CONTAINER"/>
</classpath>
```

```
<projectDescription>
  <name>pacemaker-console-maven</name>
  <projects/>
  <buildSpec>
    <buildCommand>
      <name>org.eclipse.jdt.core.javabuilder</name>
    </buildCommand>
  </buildSpec>
  <natures>
    <nature>org.eclipse.jdt.core.javanature</nature>
  </natures>
</projectDescription>
```

# Eclipse Variable Definition



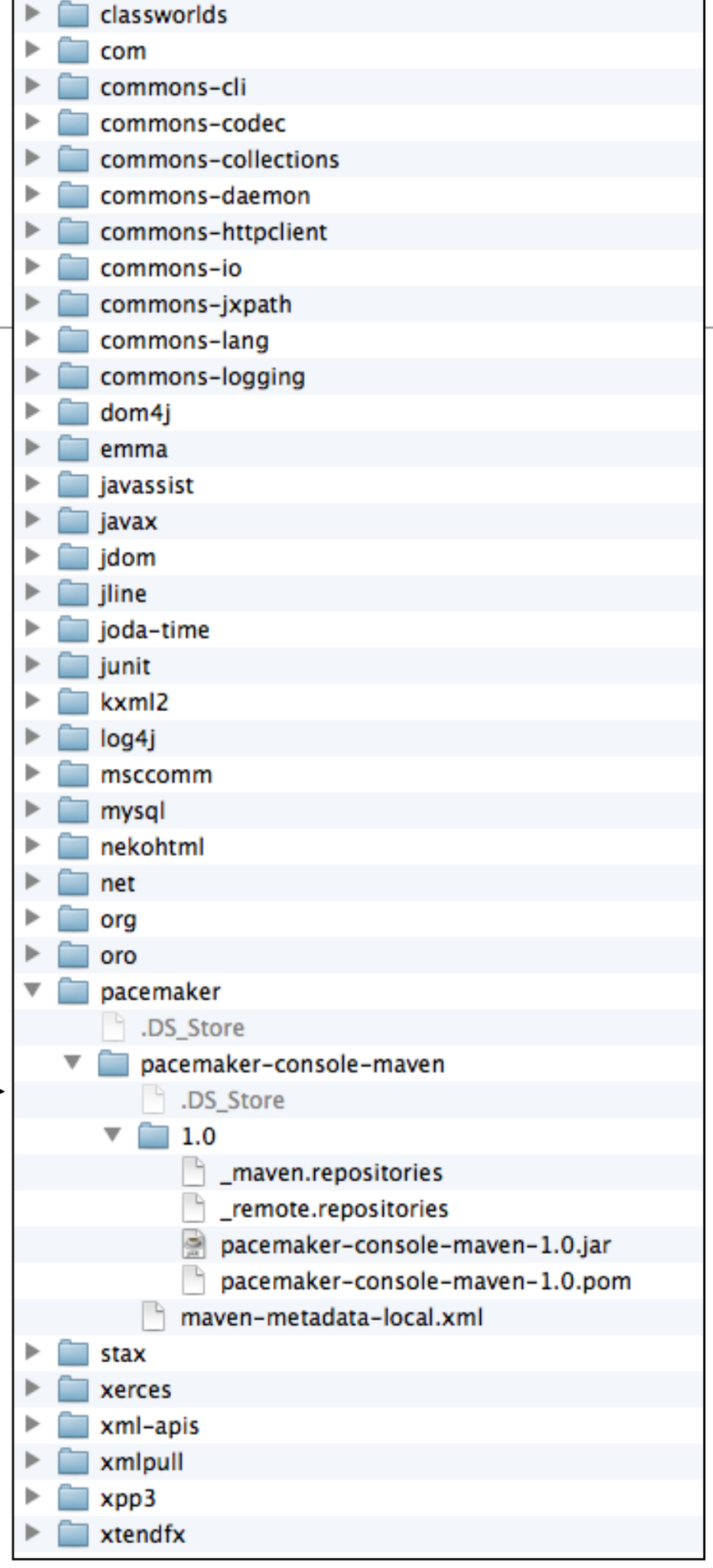



# Install

- Copies the generated package to the local Maven repository

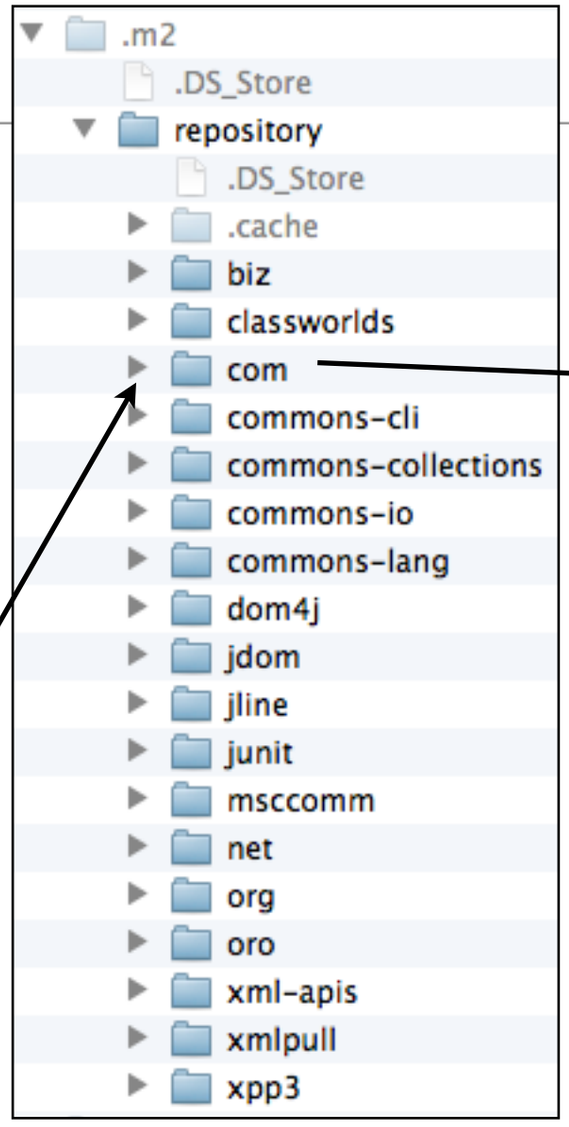
```
mvn install
```

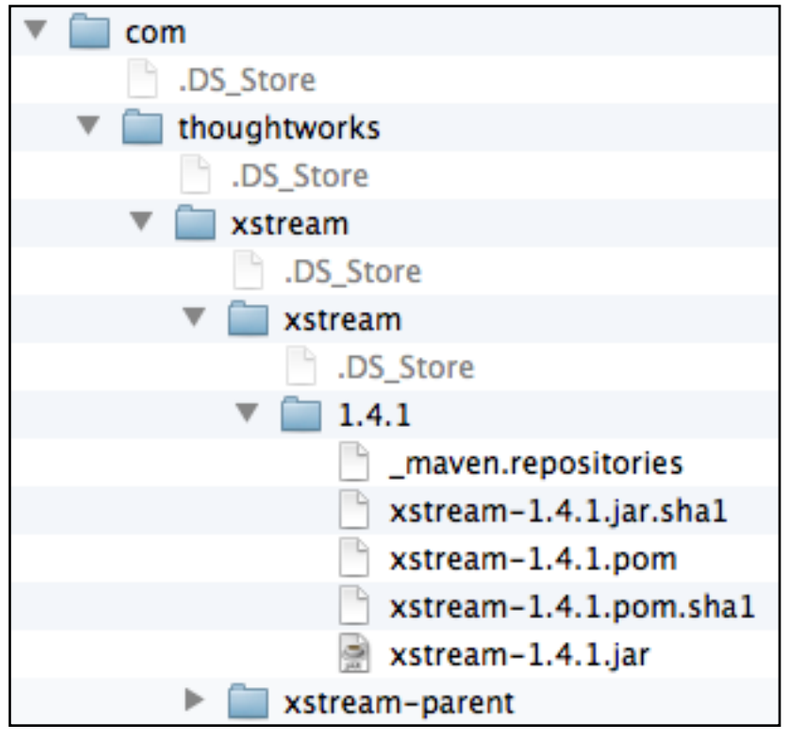




# Dependencies also in Repositories

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <version>15.0</version>
  </dependency>

  <dependency>
    <groupId>com.thoughtworks.xstream</groupId>
    <artifactId>xstream</artifactId>
    <version>1.4.4</version>
  </dependency>

  <dependency>
    <groupId>asg-cliche</groupId>
    <artifactId>asg-cliche</artifactId>
    <version>1.0</version>
  </dependency>
</dependencies>
```



# Launch Script - Unix

- Maven Repository
- Classpath for pacemaker
- Launch Command

```
M2_REPO=/Users/edeleastar/.m2/repository

export CLASSPATH=%CLASSPATH%:\
${M2_REPO}/pacemaker/pacemaker-console-maven/1.0/pacemaker-console-maven-1.0.jar:\
${M2_REPO}/com/thoughtworks/xstream/xstream/1.4.4/xstream-1.4.4.jar:\
${M2_REPO}/com/google/guava/guava/15.0/guava.15.0.jar:\
${M2_REPO}/asg-cliche/asg-cliche/1.0/asg-cliche-1.0.jar

java controllers.Main
```

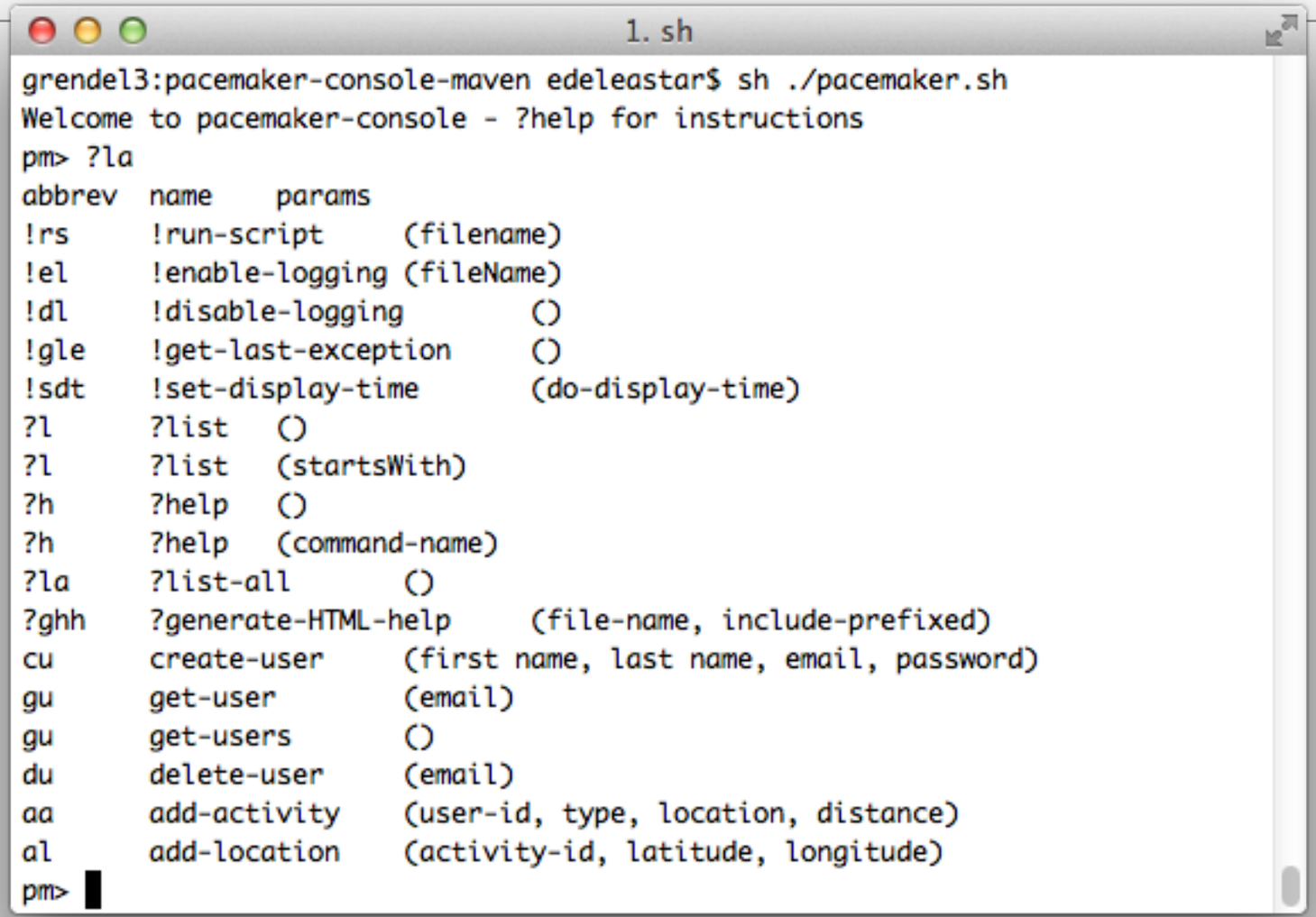
# Launch Script - Windows

- Maven Repository
- Classpath for pacemaker
- Launch Command

```
echo off
set M2_REPO=C:/docume1/eamonn1/.m2/repository

set CLASSPATH=
call cpappend.bat "%M2_REPO%/pacemaker/pacemaker-console-maven/1.0/pacemaker-console-maven-1.0.jar;"
call cpappend.bat "%M2_REPO%/com/thoughtworks/xstream/xstream/1.4.4/xstream-1.4.4.jar;"
call cpappend.bat "%M2_REPO%/com/google/guava/guava/15.0/guava.15.0.jar;"
call cpappend.bat "%M2_REPO%/asg-cliche/asg-cliche/1.0/asg-cliche-1.0.jar;"

java controllers.Main
```

```
set CLASSPATH=%CLASSPATH%;%~f1
```

cppappend.bat


1. sh
grendel3:pacemaker-console-maven edeleastar$ sh ./pacemaker.sh
Welcome to pacemaker-console - ?help for instructions
pm> ?la
abbrev  name    params
!rs     !run-script     (filename)
!el     !enable-logging (fileName)
!dl     !disable-logging        ()
!gle    !get-last-exception     ()
!sdt    !set-display-time       (do-display-time)
?l      ?list   ()
?l      ?list   (startsWith)
?h      ?help   ()
?h      ?help   (command-name)
?la     ?list-all       ()
?ghh    ?generate-HTML-help     (file-name, include-prefixed)
cu      create-user     (first name, last name, email, password)
gu      get-user        (email)
gu      get-users       ()
du      delete-user     (email)
aa      add-activity    (user-id, type, location, distance)
al      add-location    (activity-id, latitude, longitude)
pm>

# maven-android-plugin

(renamed to android-maven-plugin) Easy to use Maven plugin for Android™!





## Documentation

*Documentation collection for Android Maven Plugin*

Featured

# Maven: The Complete Reference - Android chapter

Documentation in book form is available with the Android chapter of the free book Maven: The complete R
http://www.sonatype.com/books/mvnref-book/reference/android-dev.html

# Wiki Documentation

## Start here

- GettingStarted quickly!
- See some Samples of how to use Android Maven Plugin
- Tips and tricks for using the plugin even easier!

## Eclipse

- EclipseIntegration - help collecti
- QuickStartForEclipseProject

## Reference

- Maven Site - Auto-generated doc
  - Plugin goals reference - All
- See AddOns for Building for And
- See ApkLib for how to bring in o
- **Signing**
  - See SigningAPKWithMaven
  - See ZipalignAPKBuiltByMA

## See also

- Take part in the mailing list Mav

*Comment by XXXX, May 2, 2013*

*I found the Maven android plugin on several plugin collection page, so I thought this will be a resource I should use, but I gave up, because I didn't even found out what is this in a few minutes. You should really write at least one paragraph about this project for the newbies like me. I am not too lazy to put time in learning, but I will not spend time for something which has no real documentation, and I will presume there is none, if I didn't found out what is this on the project homepage.*