Quick Maven

General

Maven is a software tool for Java project management and build automation. It is similar in functionality to the Apache Ant tool, but is based on different concepts.

In this sheet a number of commands used for Overture are described. They are all terminal commands and executed in a terminal in the following form.

```
mvn option goal
```

Where a maven **goal** is the desired action which should be performed on the maven project where the terminal is currently active.

Build Lifecycles

The build lifecycle is a list of named *phases* which can be used to give order to **goal** execution. This means that a goal of a plug-in can be bound to a specific *phase* thus called when this *phase* is reached in the build:

- process-resources
- compile
- process-test-resources
- test-compile
- test
- package
- install
- deploy

Basic goals

compile	Compiles the project and all sub projects
test	Runs compile and then tests
package	Build a jar containing the compiled resources of the project.
install	Installs the jar of the project in the local repository including it's pom so other plug-ins which depends on it just can add this to their classpath.
assembly:assembly	Build an executable jar including all depended plug-ins.[a]

Options

−X	Show debug info
−o	Run off line. This speeds up the process a lot if you already have downloaded all depended plug-ins to the local repository.
−N	Do not run recursive. (Do not run sub projects)
−fae	Fail at end. This is use full if running a root project. Allow all non-impacted builds to complete.
−Dmaven.test.skip=true	Skip all testing

---

[a]Not a default goal it comes from a assembly plug-in where the manifest needs to be specified

Eclipse

maven-eclipse-plugin Goals

eclipse:eclipse	Creates Eclipse project files for a project.
eclipse:clean	Deletes all Eclipse files from a project.

maven-psteclipse-plugin Goals

psteclipse:eclipse-plugin	Generates Eclipse files based on the packing of the pom:	
	source-plugin	Creates the Eclipse structure if missing.
	binary-plugin	Creates the Eclipse structure if missing and a lib folder with the jars from all depended jar files. It also generated the Manifest with all classes from the jar files exported.[a]

---

[a]It can also be made to import a number of packages.