

Basics

<code>mvn <plugin>:<goal> [-Doption1 -Doption2 ...]</code>	Basic maven invocation of <goal> (multiple goals possible)
<code>mvn <phase> [-Doption1 -Doption2 ...]</code>	Execute all maven goals until phase <phase> (multiple phases possible)
<code>mvn -h</code>	Getting help on command line parameters
<code>mvn install</code>	Standard mvn call: build the package files (jar, war, etc), run unit and integration tests, submit files to local mvn repository (this call works by default recursively).
<code>mvn -o</code>	Working offline
<code>mvn -U</code>	Ignore earlier failures to download latest snapshots (sometimes useful when mvn does not find a snapshot that seems to be present)
<code>mvn -N</code>	Skip visiting of child artifacts (the given goal or phase will be executed only for the current artifact - no recursive descent)
<code>mvn -DtestFailureIgnore=true <goal></code>	To continue the maven build even if a test fails
<code>mvn install -DskipTests=true</code>	To make maven install without launching the tests (but it generates the test-jar)
<code>mvn test</code>	To run unit tests
<code>mvn clean</code>	To clean up
<code>mvn -P<profile></code>	Activate a profile, deactivate all others
<code>mvn -P+<profile></code>	Activate a profile
<code>mvnrec <plugin>:<goal> or <phase></code>	Recursively execute the goal or the phase (like LEAF build.rec or EL4Ant? build.rec) (available from EL4J 1.3)

Important *phases* of maven (a concatenation of phases is called *lifecycle*)

- **Build** lifecycle: *validate, compile, test, package, integration-test, verify, install, deploy*
- **Clean** lifecycle: *clean*
- **Site** lifecycle: *site, site-deploy*

Configuration

Environment variable	Significance
M2_HOME	Root of where the <code>mvn</code> executable is located
MAVEN_OPTS	Command line options of the JVM launching maven
M2_REPO	Location of the local maven repository

Configuration file for maven: `~/.m2/settings.xml`.

We recommend to set the following parameter on the command line of maven: `-fae` "fail at end" Rationale: (1) you see explicitly what tasks have been run and (2) maven continues until the end (without breaking on intermediate errors) BTW: We added this parameter to the default maven launch script.

Eclipse Plugin

<code>mvn eclipse:eclipse</code>	To generate Eclipse project descriptor after configuring the dependencies in pom.xml (see also next point)
<code>mvn clean eclipse:eclipse -DdownloadSources=true</code>	To setup the source versions of all included libraries (for debugging and documentation convenience)
<code>mvn eclipse:configure-workspace</code>	To add the classpath variable M2_REPO to Eclipse