Basics

`mvn <plugin>:<goal> [-Doption1 -Doption2 ... ]`	Basic maven invocation of <goal> (multiple goals possible)
`mvn <phase> [-Doption1 -Doption2 ... ]`	Execute all maven goals until phase <phase> (multiple phases possible)
`mvn -h`	Getting help on command line parameters
`mvn install`	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).
`mvn -o`	Working offline
`mvn -U`	Ignore earlier failures to download latest snapshots (sometimes useful when mvn does not find a snapshot that seems to be present)
`mvn -N`	Skip visiting of child artifacts (the given goal or phase will be executed only for the current artifact - no recursive descent)
`mvn -DtestFailureIgnore=true <goal>`	To continue the maven build even if a test fails
`mvn install -DskipTests=true`	To make maven install without launching the tests (but it generates the test-jar)
`mvn test`	To run unit tests
`mvn clean`	To clean up
`mvn -P<profile>`	Activate a profile, deactivate all others
`mvn -P+<profile>`	Activate a profile
`mvnrec  <plugin>:<goal>` or `<phase>`	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 `mvn` 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

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