mvn compiler:help -Ddetail=true -Dgoal=compile

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building 15-PluginsAndGoals 1.0

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- maven-compiler-plugin:3.1:help (default-cli) @ 15-PluginsAndGoals ---

[INFO] Maven Compiler Plugin 3.1

The Compiler Plugin is used to compile the sources of your project.

compiler:compile

Compiles application sources

Available parameters:

annotationProcessors

Names of annotation processors to run. Only applies to JDK 1.6+ If not

set, the default annotation processors discovery process applies.

compilerArgs

Sets the arguments to be passed to the compiler if fork is set to true.

Example:

<compilerArgs>

<arg>-Xmaxerrs=1000</arg>

<arg>-Xlint</arg>

</compilerArgs>

compilerArgument

Sets the unformatted single argument string to be passed to the compiler

if fork is set to true. To pass multiple arguments such as -Xmaxerrs 1000

(which are actually two arguments) you have to use compilerArguments.

This is because the list of valid arguments passed to a Java compiler

varies based on the compiler version.

compilerArguments

Sets the arguments to be passed to the compiler (prepending a dash) if

fork is set to true.

This is because the list of valid arguments passed to a Java compiler

varies based on the compiler version.

To pass -Xmaxerrs 1000 -Xlint -Xlint:-path -Averbose=true you should

include the following:

<compilerArguments>

<Xmaxerrs>1000</Xmaxerrs>

<Xlint/>

<Xlint:-path/>

<Averbose>true</Averbose>

</compilerArguments>

compilerId

The compiler id of the compiler to use. See this guide for more

information.

compilerReuseStrategy

Strategy to re use javacc class created:

- reuseCreated (default): will reuse already created but in case of

multi-threaded builds, each thread will have its own instance

- reuseSame: the same Javacc class will be used for each compilation even

for multi-threaded build

- alwaysNew: a new Javacc class will be created for each compilation

Note this parameter value depends on the os/jdk you are using, but the

default value should work on most of env.

compilerVersion

Version of the compiler to use, ex. '1.3', '1.5', if fork is set to true.

debug

Set to true to include debugging information in the compiled class files.

debuglevel

Keyword list to be appended to the -g command-line switch. Legal values

are none or a comma-separated list of the following keywords: lines, vars,

and source. If debug level is not specified, by default, nothing will be

appended to -g. If debug is not turned on, this attribute will be ignored.

encoding

The -encoding argument for the Java compiler.

excludes

A list of exclusion filters for the compiler.

executable

Sets the executable of the compiler to use when fork is true.

failOnError

Indicates whether the build will continue even if there are compilation

errors.

fileExtensions

file extensions to check timestamp for incremental build default contains

only .class

forceJavacCompilerUse

compiler can now use javax.tools if available in your current jdk, you can

disable this feature using -Dmaven.compiler.forceJavacCompilerUse=true or

in the plugin configuration

fork

Allows running the compiler in a separate process. If false it uses the

built in compiler, while if true it will use an executable.

generatedSourcesDirectory

Specify where to place generated source files created by annotation

processing. Only applies to JDK 1.6+

includes

A list of inclusion filters for the compiler.

maxmem

Sets the maximum size, in megabytes, of the memory allocation pool, ex.

'128', '128m' if fork is set to true.

meminitial

Initial size, in megabytes, of the memory allocation pool, ex. '64', '64m'

if fork is set to true.

mojoExecution

optimize

Set to true to optimize the compiled code using the compiler's

optimization methods.

outputFileName

Sets the name of the output file when compiling a set of sources to a

single file. expression='${project.build.finalName}'

proc

Sets whether annotation processing is performed or not. Only applies to

JDK 1.6+ If not set, both compilation and annotation processing are

performed at the same time.

Allowed values are:

- none - no annotation processing is performed.

- only - only annotation processing is done, no compilation.

showDeprecation

Sets whether to show source locations where deprecated APIs are used.

showWarnings

Set to true to show compilation warnings.

skipMain

Set this to 'true' to bypass compilation of main sources. Its use is NOT

RECOMMENDED, but quite convenient on occasion.

skipMultiThreadWarning

source

The -source argument for the Java compiler.

staleMillis

Sets the granularity in milliseconds of the last modification date for

testing whether a source needs recompilation.

target

The -target argument for the Java compiler.

useIncrementalCompilation

to enable/disable incrementation compilation feature

verbose

Set to true to show messages about what the compiler is doing.

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 1.671 s

[INFO] Finished at: 2017-11-09T23:38:01Z

[INFO] Final Memory: 7M/155M

[INFO] ------------------------------------------------------------------------