mvn help:describe -Dcmd=compiler:

compile -Ddetail

[INFO] Scanning for projects...

[INFO]

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

[INFO] Building 16-PluginProperties 1.0

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

[INFO]

[INFO] --- maven-help-plugin:2.2:describe (default-cli) @ 16-PluginProperties ---

[INFO] 'compiler:compile' is a plugin goal (aka mojo).

Mojo: 'compiler:compile'

compiler:compile

Description: Compiles application sources

Implementation: org.apache.maven.plugin.compiler.CompilerMojo

Language: java

Bound to phase: compile

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>

Deprecated. use {@link #compilerArgs} instead.

compilerId (Default: javac)

User property: maven.compiler.compilerId

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

information.

compilerReuseStrategy (Default: ${reuseCreated})

User property: maven.compiler.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

User property: maven.compiler.compilerVersion

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

debug (Default: true)

User property: maven.compiler.debug

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

debuglevel

User property: maven.compiler.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 (Default: ${project.build.sourceEncoding})

User property: encoding

The -encoding argument for the Java compiler.

excludes

A list of exclusion filters for the compiler.

executable

User property: maven.compiler.executable

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

failOnError (Default: true)

User property: maven.compiler.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 (Default: false)

User property: maven.compiler.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 (Default: false)

User property: maven.compiler.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 (Default:

${project.build.directory}/generated-sources/annotations)

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

User property: maven.compiler.maxmem

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

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

meminitial

User property: maven.compiler.meminitial

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

'64m' if fork is set to true.

mojoExecution

User property: mojoExecution

(no description available)

optimize (Default: false)

User property: maven.compiler.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 (Default: false)

User property: maven.compiler.showDeprecation

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

showWarnings (Default: false)

User property: maven.compiler.showWarnings

Set to true to show compilation warnings.

skipMain

User property: maven.main.skip

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

RECOMMENDED, but quite convenient on occasion.

skipMultiThreadWarning (Default: false)

User property: maven.compiler.skipMultiThreadWarning

(no description available)

source (Default: 1.5)

User property: maven.compiler.source

The -source argument for the Java compiler.

staleMillis (Default: 0)

User property: lastModGranularityMs

Sets the granularity in milliseconds of the last modification date for

testing whether a source needs recompilation.

target (Default: 1.5)

User property: maven.compiler.target

The -target argument for the Java compiler.

useIncrementalCompilation (Default: true)

User property: maven.compiler.useIncrementalCompilation

to enable/disable incrementation compilation feature

verbose (Default: false)

User property: maven.compiler.verbose

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

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

[INFO] BUILD SUCCESS

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

[INFO] Total time: 4.609 s

[INFO] Finished at: 2017-11-10T07:18:09Z

[INFO] Final Memory: 10M/123M

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