## 构建maven工程，报错

CoreException: Could not get the value for parameter compilerId for plugin execution default-compile:PluginResolutionException:Pluginorg.apache.maven .plugins:maven-compiler-plugin:3.1 or one of its dependencies could not be resolved: The following artifacts could not be resolved: com.google.code.findbugs:jsr305:jar:2.0.1, org.codehaus.plexus:plexus-compiler-

**解决方案：**

在Eclipse中使用Alt+F5快捷键，在弹出的Update Maven Project对话框中选择报错的Maven工程，勾选下图中的Force Update of Snapshots/Releases，更新结束后，问题解决。

## [Log4j日志等级](http://blog.java1234.com/blog/articles/268.html)

**说明：**Log4j根据日志信息的重要程度，分OFF、FATAL、ERROR、WARN、INFO、DEBUG、ALL

当然再细分的话 还有FATAL（严重错误），

**Log4j建议只使用四个级别**，优先级从高到低分别是ERROR、WARN、INFO、DEBUG

若是log4j.rootLogger=debug, Console ,File ； 则会输出ERROR、WARN、INFO、DEBUG

若是log4j.rootLogger=INFO, Console ,File ； 则会输出ERROR、WARN、INFO

## [Log4j rootLogger配置](http://blog.java1234.com/blog/articles/269.html)

**说明：**Log4j 根配置语法

|  |  |
| --- | --- |
| 1 | log4j.rootLogger = [ level ] , appenderName, appenderName |

指代 把指定级别的日志信息输出到指定的一个或者多个位置

比如：

log4j.rootLogger=debug, Console ,File ,DailyRollingFile ,RollingFile, DFile, EFILE

## [Log4j appender输出类型配置](http://blog.java1234.com/blog/articles/270.html)

**说明：**Log4j官方的appender给出了一下几种实现

org.apache.log4j.ConsoleAppender（控制台），

org.apache.log4j.FileAppender（文件），

org.apache.log4j.DailyRollingFileAppender（每天产生一个日志文件），

org.apache.log4j.RollingFileAppender（文件大小到达指定尺寸的时候产生一个新的文件）

比如：

log4j.rootLogger=DEBUG, Console ,File ,DailyRollingFile ,RollingFile

**#Console**

log4j.appender.Console=org.apache.log4j.ConsoleAppender

log4j.appender.Console.layout=org.apache.log4j.PatternLayout

log4j.appender.Console.layout.ConversionPattern=%d [%t] %-5p [%c] - %m%n

**#File**

log4j.appender.File = org.apache.log4j.FileAppender

log4j.appender.File.File = C://log2.log

log4j.appender.File.layout = org.apache.log4j.PatternLayout

log4j.appender.File.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n

**#DailyRollingFile**

log4j.appender.DailyRollingFile = org.apache.log4j.DailyRollingFileAppender

log4j.appender.DailyRollingFile.File = C://log3.log

log4j.appender.DailyRollingFile.layout = org.apache.log4j.PatternLayout

log4j.appender.DailyRollingFile.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n

**#RollingFile**

log4j.appender.RollingFile = org.apache.log4j.RollingFileAppender

log4j.appender.RollingFile.File = C://log4.log

log4j.appender.RollingFile.MaxFileSize=1KB

log4j.appender.RollingFile.MaxBackupIndex=3

log4j.appender.RollingFile.layout = org.apache.log4j.PatternLayout

log4j.appender.RollingFile.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n

## [log4j layout日志信息格式](http://blog.java1234.com/blog/articles/271.html)

**说明：PatternLayout是我们以后推荐使用的，很灵活**；

有个ConversionPattern属性，灵活配置输出属性：

%m 输出代码中指定的消息；

%M 输出打印该条日志的方法名；

**%p 输出优先级**，即DEBUG，[INFO](http://baike.baidu.com/item/INFO" \t "http://blog.java1234.com/blog/articles/_blank)，WARN，ERROR，FATAL；

%r 输出自应用启动到输出该log信息耗费的毫秒数；

**%c 输出所属的类目**，通常就是所在类的全名；

%t 输出产生该日志事件的线程名；

**%n 输出一个回车换行符**，Windows平台为"rn”，Unix平台为"n”；

**%d 输出日志时间点的日期或时间**，默认格式为ISO8601，也可以在其后指定格式，比如：%d{yyyy-MM-dd HH:mm:ss,SSS}，输出类似：2002-10-18 22:10:28,921；

**%l 输出日志事件的发生位置**，及在代码中的行数；

## [Log4j Threshold属性指定输出等级](http://blog.java1234.com/blog/articles/272.html)

**说明：**

log4j.rootLogger=DEBUG, Console ,DFile ,EFILE

**#Console**

log4j.appender.Console=org.apache.log4j.ConsoleAppender

log4j.appender.Console.layout=org.apache.log4j.TTCCLayout

**#DEBUGFile**

log4j.appender.DFile = org.apache.log4j.FileAppender

log4j.appender.DFile.File = C://DEBUG.log

log4j.appender.DFile.layout = org.apache.log4j.PatternLayout

log4j.appender.DFile.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n

log4j.appender.DFile.Threshold = DEBUG

**#ERRORFile**

log4j.appender.EFILE = org.apache.log4j.FileAppender

log4j.appender.EFILE.File = C://ERROR.log

log4j.appender.EFILE.layout = org.apache.log4j.PatternLayout

log4j.appender.EFILE.layout.ConversionPattern =%d [%t] %-5p [%c] - %m%n

log4j.appender.EFILE.Threshold = ERROR

rootLogger里配置DEBUG，

**DFile的Threshold 配置为DEBUG，会输出所有信息**

**EFILE的Threshold 配置为ERROR 只输入ERROR信息**；