

Maven有个[IRC交流频道](#) `#maven`

## 1. 命令行

### 1.1 -U,--update-snapshots

[参考](#)

本地仓库中的 `**lastUpdated` 文件就和这个命令有关, `**lastUpdated` 是个时间戳文件, 记录了最新更新时间戳和仓库, maven会根据xml配置文件中的 `<repository>` 标签下的 `<updatePolicy>` 更新策略检测jar是否更新。

此参数可以强制进行更新检查;

### 1.2 -t, --toolchains

[参考](#)

### 1.3 -T,--threads

[参考](#)

例如: 在4核8线程的CPU上 (i7-8850u)

```
-T 0.5C
```

```
Using the MultiThreadedBuilder implementation with a thread count of 4
```

```
-T 1.5C
```

```
Using the MultiThreadedBuilder implementation with a thread count of 12
```

### 1.4 -s,--settings

设置用户setting.xml文件位置, 默认是在 `~/.m2/setting.xml` ;

### 1.5 -rf,--resume-from

表示: 从构建(模块)列表中的某个模块开始顺序构建, 前面的就不再执行构建

例如: [参考](#)

### 1.6 -q,--quiet

静默模式, 除了错误, 什么都不提示;

### 1.7 -pl,--projects <moduleName>

构建特定的模块, modulesNames使用逗号分隔;

### 1.8 -P,--activate-profiles <arg>

设置激活的profiles, 逗号分隔;

## 1.9 -o, --offline

离线模式, [参考](#)

## 1.10 -nsu,--no-snapshot-updates

禁止snapshot更新;

## 1.11 -N,--non-recursive

禁止递归, 通俗讲就是不构建子模块; [参考](#)

## 1.12 -lR,--legacy-local-repository

作用, 不知!

## 1.13 -l,--log-file <arg>

日志文件位置;

## 1.14 -gs,--global-settings <arg>

全局配置文件位置;

## 1.15 -f,--file <arg>

设置pom.xml文件位置;

## 1.16 -e, --errors

设置日志级别为error, 打印的信息更加详细

```
Caused by: org.codehaus.plexus.util.xml.pull.XmlPullParserException: Unrecognised tag: 'groupId'
(position: START_TAG seen ...<plugin>\r\n\t\t\t\t<groupId>... @49:16)
    at org.apache.maven.model.io.xpp3.MavenXpp3ReaderEx.checkUnknownElement (MavenXpp3ReaderEx.java:180)
    at org.apache.maven.model.io.xpp3.MavenXpp3ReaderEx.parsePlugin (MavenXpp3ReaderEx.java:3141)
    at org.apache.maven.model.io.xpp3.MavenXpp3ReaderEx.parseBuild (MavenXpp3ReaderEx.java:1025)
```

## 1.17 -D,--define <arg>

定义一个系统属性, 比如: mvn -DapplicationName=testDemo

## 1.18 -b,--builder <arg>

设置构建策略; 参数为builder的id;

## 1.19 -B,--batch-mode

在非交互(批处理)模式下运行(该模式下,当Maven需要输入时,它不会停下来接受用户的输入,而是使用合理的默认值);

## 1.20 -X, --debug

debug模式, 此模式下, 控制台会打印更加详细的日志, 比如配置文件的读取顺序:

```
[DEBUG] Populating class realm maven.api
[INFO] Error stacktraces are turned on.
```

```
[DEBUG] Message scheme: color
[DEBUG] Message styles: debug info warning error success failure strong mojo project
[DEBUG] Reading global settings from C:\Tools\apache-maven-3.5.3\bin\..\conf\settings.xml
[DEBUG] Reading user settings from C:\Users\starzonecn\.m2\settings.xml
[DEBUG] Reading global toolchains from C:\Tools\apache-maven-3.5.3\bin\..\conf\toolchains.xml
[DEBUG] Reading user toolchains from C:\Users\starzonecn\.m2\toolchains.xml
[DEBUG] Using local repository at D:\Repository-maven
[DEBUG] Using manager EnhancedLocalRepositoryManager with priority 10.0 for D:\Repository-maven
```

## 了解

- `-fn,--fail-never`
- `-ff,--fail-fast`
- `-fae,--fail-at-end`

## 2. 插件

### 2.1 maven-resources-plugin

[博客](#)

### 2.2 help

命令:

- `mvn help:effective-settings` - 查看当前有效的setting配置;
- `mvn help:system` - 打印环境变量和java系统属性;
- `mvn help:effective-pom` - 打印有效的pom, 包括父pom;
- `mvn help:active-profiles` - 显示当前激活的profiles;

## 3. 插件开发

### 3.1 插件缩写

场景: 比如maven-compile-plugin插件, 只需要执行 `mvn compile:compile`, 不必添加groupId和完整的artifactID - [参考](#)

#### 3.1.1 maven默认方式

By default, Maven will make a guess at the plugin-prefix to be used, by removing any instances of "maven" or "plugin" surrounded by hyphens in the plugin's artifact ID.

maven默认是去除所有连字符包围的 `maven` 和 `plugin` 单词, 但是有些要求:

- `maven-${prefix}-plugin` - for official (官方) plugins maintained by the Apache Maven team itself (you **must not** use this naming pattern for your plugin, see [this note for more informations](#))
- `${prefix}-maven-plugin` - for plugins from other sources

使用 `maven-<yourplugin>-plugin` 形式, 便侵犯了Apache的商标:

Calling it `maven-<yourplugin>-plugin` (note "Maven" is at the beginning of the plugin name) is **strongly discouraged** since it's a **reserved naming pattern for official Apache Maven plugins maintained by the Apache Maven team** with groupId `org.apache.maven.plugins`. Using this naming pattern is an infringement of the Apache Maven Trademark.

#### 3.1.2 直接定义

```

<project>
...
<build>
...
<plugins>
...
<plugin>
<artifactId>maven-plugin-plugin</artifactId>
<version>2.3</version>
<configuration>
...
<goalPrefix>somePrefix</goalPrefix>
</configuration>
</plugin>
</plugins>
</build>
</project>

```

### 3.1.3 配置

最后要在settings.xml中添加配置:

```

<pluginGroups>
  <pluginGroup>sample.plugin</pluginGroup>
</pluginGroups>

```

## 3.2 插件属性

```

package com.example.demo.multi.springBoot.mojo;

import org.apache.commons.lang3.StringUtils;
import org.apache.maven.plugin.AbstractMojo;
import org.apache.maven.plugin.MojoExecutionException;
import org.apache.maven.plugin.MojoFailureException;
import org.apache.maven.plugins.annotations.LifecyclePhase;
import org.apache.maven.plugins.annotations.Mojo;
import org.apache.maven.plugins.annotations.Parameter;

/**
 * @author: LiHongxing
 * @email: lihongxing@bluemoon.com.cn
 * @date: Create in 2019/1/3 14:35
 * @modified:
 */
@Mojo(name = "printPc", defaultPhase = LifecyclePhase.COMPILE)
public class GreetingMojo extends AbstractMojo {

    /* defaultValue是String类型, 但是maven会自动转换类型 */
    @Parameter(property = "printPc.printTip", defaultValue = "true")
    private boolean printTip;
    @Parameter(property = "printPc.projectVersion",
        defaultValue = "${project.version}")
    private String projectVersion;

    @Override
    public void execute() throws MojoExecutionException, MojoFailureException {
        if (this.printTip) {
            getLog().info("执行Greeting插件。。。");
        }
        getLog().info( getPluginContext().toString() );
        // StringUtils是common-lang3的包
        if (StringUtils.isNotBlank(this.projectVersion)){
            getLog().info("projectVersion = " + projectVersion);
        }
        if (this.printTip) {
            getLog().info("-----");
        }
    }
}

```

```
}  
}
```

- `@Parameter` 注解的 `defaultValue` 属性可以通过 `${表达式}` 获取项目的属性，如上面代码中的 `${project.version}`，项目变量列表：

### 3.3 更多

参考[blog](#);

## 4. 概念

---

### 3.1 SNAPSHOT

如果在项目配置文件中（无论是build.gradle还是pom.xml）指定的版本号带有'-SNAPSHOT'后缀，比如版本号为'JUnit4.10-SNAPSHOT'，那么打出的包就是一个快照版本。

#### 3.1.1 查找频率

分为4种频率：`always`、`daily`、`interval`、`never`

- `daily` - 默认值
- `interval` - 单位是分钟

```
<repository>  
  <id>myRepository</id>  
  <url>...</url>  
  <snapshots>  
    <enabled>true</enabled>  
    <!-- 更新频率策略 -->  
    <updatePolicy>XXX</updatePolicy>  
  </snapshots>  
</repository>
```

### 扩展阅读

---

## Apache License

// [TODO](#)