第1章 mavne command

作者：Jeffen

更新日期：2015-10-17

# Build Tool 构建工具/包管理

## 构件(Artifact)

## 坐标（Coordinate）

groupId

artifactId

Version

Packaging

## 依赖（Dependency）

## POM

解耦/正交性 decouple

反应堆（Reactor）

Profile

## 约定优于配置（Convention Over Configuration）

源码目录：src/main/java/

编译输出目录：target/classes/

# 常用命令

## 创建命令

**可以在执行目录下产生一个创建的项目目录**

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn archetype:generate -DarchetypeCatalog=Internal

选择16，输入groupid等，将产生一个新项目

## 创建eclipse项目

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn eclipse:eclipse

在eclipse中可以通过import可以导入项目，导入成功后，在eclipse的项目中右键弹出菜单，选择Maven>Enable dependency management

## 创建idea项目

参见[文章](http://maven.apache.org/plugins/maven-idea-plugin/)

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn idea:idea

## 修改jdk版本

（1）全局修改conf\settings.xml

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<profiles>**
2. **<profile>**
3. **<id>**jdk-1.6**</id>**
4. **<activation>**
5. **<activeByDefault>**true**</activeByDefault>**
6. **<jdk>**1.6**</jdk>**
7. **</activation>**
8. **<properties>**
9. **<maven.compiler.source>**1.6**</maven.compiler.source>**
10. **<maven.compiler.target>**1.6**</maven.compiler.target>**
11. **<maven.compiler.compilerVersion>**1.6**</maven.compiler.compilerVersion>**
12. **</properties>**
13. **</profile>**
14. **</profiles>**

（2）pom修改

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<plugin>**
2. **<groupId>**org.apache.maven.plugins**</groupId>**
3. **<artifactId>**maven-compiler-plugin**</artifactId>**
4. **<version>**2.5.1**</version>**
5. **<configuration>**
6. **<source>**1.6**</source>**
7. **<target>**1.6**</target>**
8. **<encoding>**UTF-8**</encoding>**
9. **</configuration>**
10. **</plugin>**

## cxf产生客户端代码

参见[文章](http://cxf.apache.org/docs/maven-cxf-codegen-plugin-wsdl-to-java.html)

修改pom.xml

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<plugin>**
2. **<groupId>**org.apache.cxf**</groupId>**
3. **<artifactId>**cxf-codegen-plugin**</artifactId>**
4. **<version>**2.6.1**</version>**
5. **<executions>**
6. **<execution>**
7. **<id>**generatecxfclientsources**</id>**
8. **<phase>**generate-sources**</phase>**
9. **<configuration>**
10. **<sourceRoot>**${project.build.directory}/generated/cxf**</sourceRoot>**
11. **<wsdlOptions>**
12. **<wsdlOption>**
13. **<wsdl>**http://127.0.0.1:8080/getInfoService?wsdl**</wsdl>**
14. **</wsdlOption>**
15. **</wsdlOptions>**
16. **</configuration>**
17. **<goals>**
18. **<goal>**wsdl2java**</goal>**
19. **</goals>**
20. **</execution>**
21. **</executions>**
22. **</plugin>**

运行

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn generate-sources

## 跳过单元测试

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn package -Dmaven.test.skip=true

或者

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. <plugin>
2. <groupId>org.apache.maven.plugins</groupId>
3. <artifactId>maven-surefire-plugin</artifactId>
4. <configuration>
5. <skip>true</skip>
6. </configuration>
7. </plugin>

## 查看项目依赖情况

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn  dependency:list

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn  dependency:tree

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn  dependency:analyze

## pom中使用变量

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<properties>**
2. **<test.version>**2.6**</test.version>**
3. **</properties>**
4. **<dependencies>**
6. **<dependency>**
7. **<groupId>**javax.servlet**</groupId>**
8. **<artifactId>**servlet-api**</artifactId>**
9. **<version>**${test.version}**</version>**
10. **<scope>**provided**</scope>**
11. **</dependency>**
12. **</dependencies>**

## dependency依赖范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **依赖范围(Scope)** | **对编译classpath有效** | **对测试classpath有效** | **对运行classpath有效** | **例子** |
| compile | Y | Y | Y | Spring-core |
| test |  | Y |  | JUnit |
| provided | Y | Y |  | Servelet-api |
| runtime |  | Y | Y | jdbc驱动 |
| system | Y | Y |  | 本地的，Maven仓库以外的类库文件 |

## 将一个库放到本地repository中

**[plain]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. mvn install:install-file -Dpackaging=jar -DgroupId=com.adobe.blazeds -Dversion=4.0.0.14931 -DartifactId=blazeds-common -Dfile=flex-messaging-common.jar

## 调用ant插件

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<plugin>**
2. **<groupId>**org.apache.maven.plugins**</groupId>**
3. **<artifactId>**maven-antrun-plugin**</artifactId>**
4. **<version>**1.7**</version>**
5. **<configuration>**
6. **<tasks>**
7. **<ant** antfile="build.xml"**>**
8. **<target** name="copy-all" **/>**
9. **</ant>**
10. **<ant** antfile="build.xml"**>**
11. **<target** name="copy-iroad-flex" **/>**
12. **</ant>**
13. **<property** name="mvn-classpath" refid="maven.compile.classpath"**/>**
14. **<ant** antfile="build-jar.xml"**>**
15. **<target** name="copy-all" **/>**
16. **</ant>**
17. **</tasks>**
18. **</configuration>**
19. **</plugin>**

mvn antrun:run

## 添加库到pom.xml文件中

**[html]** [view plaincopy](http://blog.csdn.net/johnnywww/article/details/7978096)

1. **<repositories>**
2. **<repository>**
3. **<id>**spring-external**</id>**
4. **<name>**Spring External Repository**</name>**
5. **<url>**http://maven.springframework.org/external**</url>**
6. **</repository>**
7. **</repositories>**