Maven安装使用简易说明文档

**作者：**

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | 版本 |  | 说明 |
| 何益武 | 1.0.0 |  | 起草初稿 |
| 何益武 | 1.0.1 |  | 增加开发指南文档，以及相应的工程案例 |

**说明：**

文档中提到的开发工具提供下载,地址:

https://www.steel56.com.cn:8443/golden/product/devshare/dev-tools

JDK等工具的基础开发环境搭建、安装配置，**参考开发指南环境篇**中的相应章节介绍：

https://www.steel56.com.cn:8443/golden/product/devshare/docs/开发指南.docx

学习maven构建管理和工程结构，可参考案例工程(WEB工程可运行)源码svn 地址：

https://www.steel56.com.cn:8443/golden/product/devshare/example/example-of-case

# 一、安装：

## 1、前提

maven运行在JDK1.4以上，安装前请验证JDK版本及环境变量.**推荐安装JDK1.6**,JDK安装及环境变量设置此处省略，具体见开发指南环境篇.

Eclipse中关于JDK的设置：

**Eclipse的设置信息，一般保存在workspace目录下，同时存在JDK1.5和JDK1.6的工程时，建议分不同的workspace进行设置，即JDK1.5一个workspace，JDK1.6一个workspace，需要切换时，切换workspace即可。**

在eclipse选项（Window🡪Preferences）中java🡪Installed JREs 设置并选择JDK1.6为默认.

设置全局编译器：java🡪compiler,选中use default compliance settings为1.6。如有多编译环境（即有其他工程是需要JDK1.5编译器的，可以在工程属性上设置，设置方法类似）

## 2、Maven安装

下载安装包解压即可

设置环境变量（**推荐设置用户环境变量**）:

MAVEN\_HOME=D:\Java\apache-maven-3.0.3

M2\_HOME=%MAVEN\_HOME%

M2\_REPO=D:\repo

(和%MAVEN\_HOME%\config\settings.xml中的localRepository值对应)

MAVEN\_OPTS= -Xms128m -Xmx512m

set PATH=%PATH%;%MAVEN\_HOME%\bin

## 3、验证Maven安装

cmd下执行：mvn–v

备注：环境变量设置或修改完毕后，需点击确定，然后在新开cmd命令窗口才能加载系统设置（操作系统的特性，诸如此类的还有很多，例如：修改系统host映射，关闭浏览器后在新开浏览器才能加载设置）

## 4、设置本地仓库位置(推荐保持默认即可):

修改文件:%M2\_HOME%\conf\settings.xml中的

localRepository: <localRepository>D:\repo</localRepository>

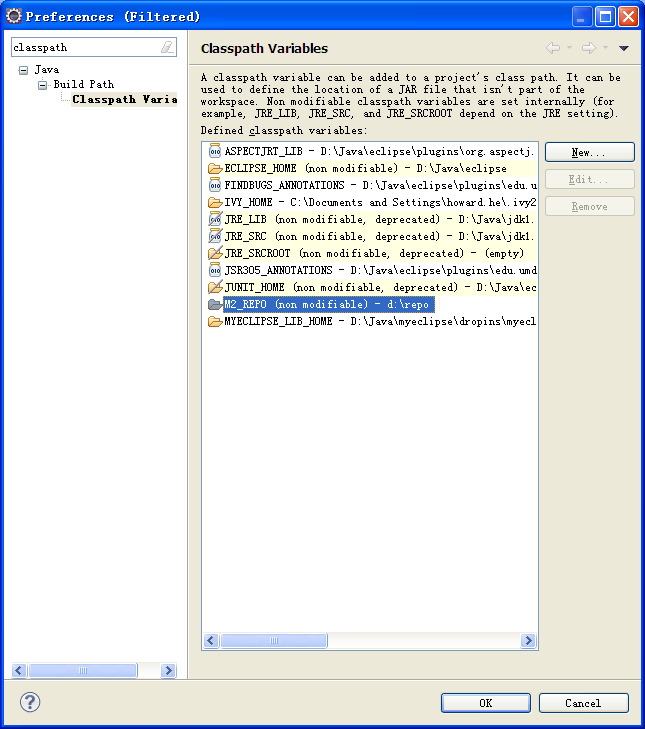
和环境变量保持一致:M2\_REPO=D:\repo

# 二、eclipse设置

## 1、设置maven变量

在eclipse设置选项(window🡪preferences)里搜索classpath,设变量 M2\_REPO=d:\repo.

如果M2\_REPO变量已存在且不能修改(如图)，是由于maven4myeclipse（myeclipse自带的maven插件）的设置的原因，修改maven4myeclipse相应的设置即可，具体设置见“2.maven插件设置”

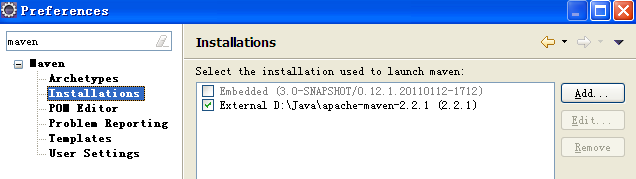


## 2、maven插件设置

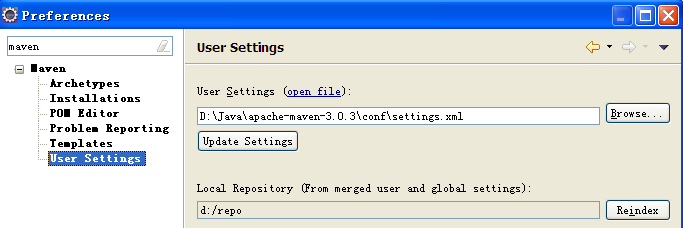
myeclipse 6以后自带maven4myeclipse插件,maven的eclipse插件还有m2eclipse，**但都不建议使用**，原因是比较耗内存，建议4G内存用户使用(**所以可跳过本小节的介绍**)  
推荐使用cmd运行maven命令，不使用eclipse插件,如使用cmd则无需以下设置,推荐安装cmdhere。  
使用cmd时的一个便捷小工具: 右键CMD  
作用是在文件或目录上，右键cmdhere可以在cmd下快速进入当前路径

maven4myeclipse插件设置:

maven的安装位置

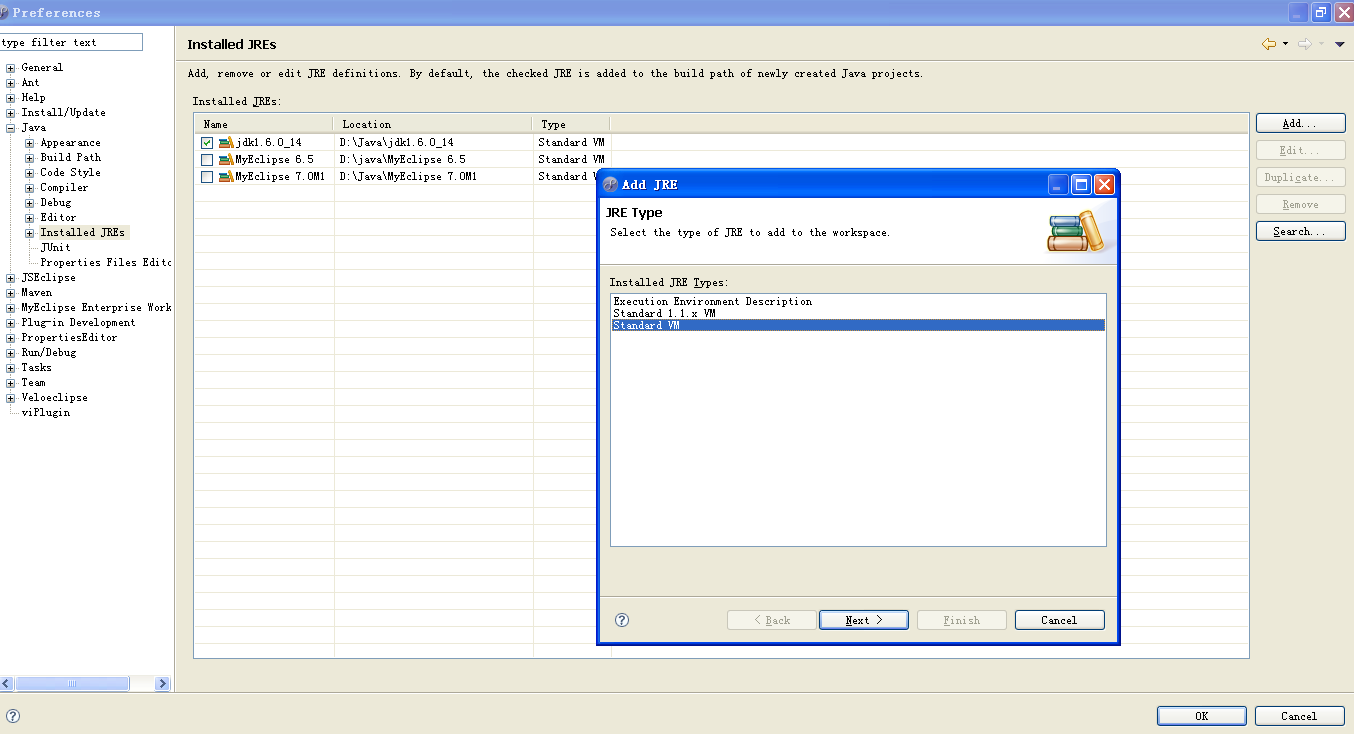


设置maven的配置文件位置，此处直接使用maven 的配置文件.如有多环境配置要求，推荐在用户目录下建立%user\_home%\.m2\settings.xml



## 3、JRE设置

由于maven运行在JDK1.4以上，所以运行maven命令前需设置eclipse的默认JRE(一般不使用eclipse或myeclipse自带的JRE)，设置如下：



# 三、私服地址映射配置

settings.xml配置文件里，私服地址均使用的是域名，没有使用IP，所以需要在系统host文件里，需加上私服的IP域名映射,例如加上：

C:\WINDOWS\system32\drivers\etc\hosts

192.168.2.210 dev.maven.com

# 四、常用操作使用说明(cmd):

## 1、svn检出工程源码

注意：不要直接从svn导入到eclipse

示例：

开发指南案例工程(WEB工程可运行)源码svn地址：

https://www.steel56.com.cn:8443/golden/product/devshare/example/example-of-case

Checkout到本地目录（**建议此目录不要在eclipse的工作目录下,工程源码路径路径建议不要有中文**）E:\work\example

## 2、eclipse导入

运行maven命令,生成eclipse工程文件和构建路径文件

示例：

cmd进入工程目录(E:\work\example)执行：eclipse.bat,其内容为：

mvneclipse:cleaneclipse:myeclipse

以上命令说明：此命令为2个命令的组合:

eclipse:clean为清除工程文件,

eclipse:myeclipse为生成JAVA WEB工程文件,eclipse:eclipse为生成JAVA 工程即java application工程

以上命令组合中，**先清除，在生成，为全量，否则为增量**,**一般推荐全量**,因为有删减等少数情况下(例如减少依赖等)，增量可能会有问题.其他命令亦如此。其他常用命令见maven命令集

上面的命令执行完之后会生成eclipse工程文件(.project)和构建路径文件(.classpath),然后就可以导入eclipse工程了.(存在模式导入：existing projects into workspace)

## 3、构建

(介绍性说明,环境搭建时可跳过)：

构建构件包，常见的有：字节码jar、源码jar包、二进制zip包，具体看pom.xml配置的build.plugin插件的配置

构建：mvn package 构建构件包;

安装:mvn clean install 构建构件包，安装到本地库，以供本地调用

## 4、发布

### 4.1、上传jar包(构件)：

**maven命令**:

运行此命令需配置账号认证信息(文档后面有介绍)：

示例：

mvndeploy:deploy–file

-DgroupId=com.skymobi.mavenapp

-DartifactId=mavenapp

-Dversion=2.0

-Dpackaging=jar

-Dfile=mavenapp-1.0-SNAPSHOT.jar -Durl=http://dev.maven.com:8081/nexus/content/repositories/thirdparty/

命令说明：-D 为jvm参数选项,前4个选项指定jar包的坐标(概念参见开发指南),如果是maven构建的jar包其中包含坐标信息(一般在jar包的META-INF/maven下),如果没有坐标信息可以自定义(可参照apache、spring、jboss等开源构件包的坐标定义)

url选项为私服库地址,**此地址需配置是否允许上传权限（开发指南相应章节有说明）**，示例地址为私服第三方库地址，还有realse(内部正式构件库)、snapshots(快照:最新构件,不稳定的构件)等库,具体参见开发指南.

手动上传jar包：具体见开发指南相应章节

### 4.2、发布工程：

即发布/上传工程包(jar、war、ear...)到maven私服，供其他开发人员外部工程调用：

注：下面的配置,一般要求工程发布或构建管理人员知晓,开发人员不需要关心

发布：mvn deploy构建、发布构件包到本地库和私服nexus库以供其他人调用使用.

要使用发布功能，pom.xml里需配置私服的发布节点，如：

<!-- 发布节点 -->

<distributionManagement>

<!-- mvn deploy -Pproduct 发布正式版本到nexus私服 -->

<repository>

<id>dev-release</id>

<name>dev-nexus-release</name>

<url>http://dev.maven.com:8081/nexus/content/repositories/releases/</url>

</repository>

<!-- mvn deploy 发布默认快照版本到nexus私服 -->

<snapshotRepository>

<id>dev-snapshot</id>

<name>dev-nexus-snapshot</name>

<url>http://dev.maven.com:8081/nexus/content/repositories/snapshots/</url>

</snapshotRepository>

</distributionManagement>

<profiles>

<!-- 生产环境,调用开发环境配置,默认激活 -->

<profile>

<id>prod</id>

<properties>

<env.profile>main</env.profile>

<project.release.version>1.0</project.release.version>

</properties>

</profile>

<!-- 开发环境,默认激活(在settings.xml里配置的) mvn deploy或mvn deploy -Pdev -->

<profile>

<id>exdev</id>

<properties>

<env.profile>main</env.profile>

<project.release.version>1.0-SNAPSHOT</project.release.version>

</properties>

</profile>

<!-- 测试环境,mvn install -Ptest或mvn deploy -Ptest -->

<profile>

<id>test</id>

<properties>

<env.profile>test</env.profile>

<project.release.version>1.0-SNAPSHOT</project.release.version>

</properties>

</profile>

</profiles>

**Setting.xml配置私服的账号认证信息（repository.id= server.id）**：

<server>

<id>dev-release</id>

<username>admin</username>

<password>golden</password>

</server>

<server>

<id>dev-snapshot</id>

<username>admin</username>

<password>golden</password>

</server>

发布时一同发布源码包和javadoc包

<build>

<plugins>

<!—源码包插件-->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-source-plugin</artifactId>

<version>2.0.2</version>

<executions>

<execution>

<id>attach-sources</id>

<goals>

<goal>jar</goal>

</goals>

</execution>

</executions>

</plugin>

<!—javadoc插件-->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-javadoc-plugin</artifactId>

<version>2.6.1</version>

<configuration>

<encoding>UTF-8</encoding>

</configuration>

</plugin>

</plugins>

</build>