# Maven简介

Maven是一个跨平台的项目管理工具。作为Apache一个开源项目，它主要服务于基于JAVA平台的项目构建、依赖管理和项目信息管理。

* 构建：清理、编译、测试、生成文档、打包和部署等都属于构建。
* 依赖管理：通过坐标系统管理第三方的类库存在的问题(版本不一致、版本冲突、依赖臃肿)
* 项目信息：管理项目信息，包括项目描述、开发者列表、版本控制系统地址、许可证、缺陷管理系统地址等。
* 提供一个免费的中央仓库，可以找到任何流行的开源类库。通过衍生工具(如Nexus)，进行快速搜索定位，Maven自动下载。

# Maven安装和配置

## 2.1 在Windows上安装Maven

**检查JDK安装**

在安装Maven之前，首先要确认正确安装了JDK。

**下载Maven**

进入Maven官网：http://maven.apache.org/

**本地安装**

将下载的压缩包解压至本地某个路径，如：E:\program\maven3.0。在系统环境变量加入M2\_HOME变量，变量值为安装目录如：E:\program\maven3.0。在PATH加入 %M2\_HOME%\bin

## 2.2 在Unix上安装Maven

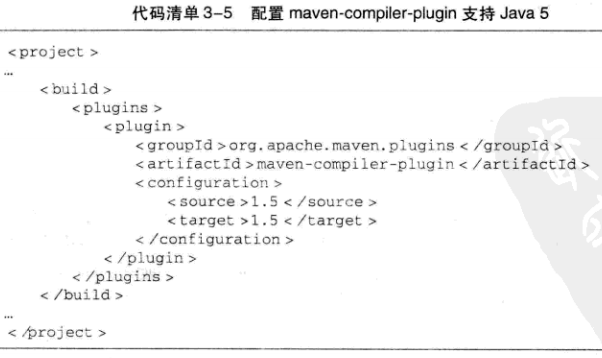
检查和下载Maven参照Windows的安装介绍。

## 2.3 安装m2eclipse

m2eclipse是eclipse下一个十分强大的Maven插件，可访问http://m2eclipse.sonatype.org/sites/m2e

通过eclipse的Help –> Install New Software菜单安装m2eclipse插件，安装完毕后重启eclipse。

Maven默认只支持JDK1.3，因此需配置更高版本的JDK。如下图：



# 3 Maven坐标和依赖

# 4 Maven资源仓库

在Maven的世界里，任何一个依赖、插件或项目构建的输出，都可称为构件。任何一个构件都有其唯一的坐标，根据坐标可以定义其在仓库中的唯一存储路径，这便是Maven的仓库布局方式。

## 4.1 仓库分类

Maven仓库分为本地仓库和远程仓库。当Maven根据坐标寻找构件的时候，首先会查看本地仓库，如存在则直接使用；如本地仓库不存在此构件或需查看是否有更新的构件版本，Maven就会去远程仓库查找，发现需要的构件后下载到本地仓库再使用。如果两个仓库都没有，则报错。

在上述分类的基础上，远程仓库又可分为中央仓库、私服、其他公共库。如图4.1所示。

私服是一种特殊的远程仓库，为了节省带宽和时间，应该在局域网内架设一个私有的仓库服务器，用其代理所有外部的远程仓库，内部的项目还能部署到私服上供其他项目使用。

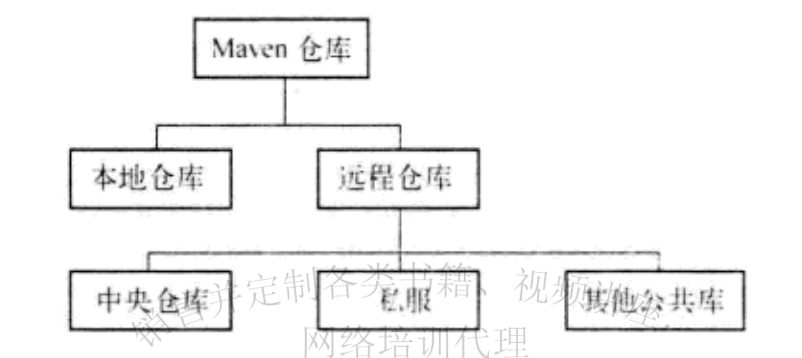


图4.1 Maven仓库的分类

## 4.2 本地仓库

默认情况下，在Windows、Linux上，每个用户在自己的用户目录下都有一个路径名为.m2/respository/的仓库目录。如用户想自定义本地仓库目录地址，可以编辑 %M2\_HOME%/conf/settings.xml文件。内容如下：

<!-- Default: ~/.m2/repository -->

<localRepository>E:\workspace\learn\repository</localRepository>

本地仓库地址修改成: E:\workspace\learn\repository

## 4.3 远程仓库

**中央仓库**

最原始的本地仓库是空的，Maven必须知道一个可用的远程仓库，才能通过执行Maven命令下载到所需的构件。中央仓库就是一个默认的远程仓库，Maven安装文件自带了中央仓库的位置。可以打开%M2\_HOME%/lib/maven-model-builder-3.0.jar访问路径org/apache/maven/model/pom-4.0.0.xml，可以看到如下配置：

<repositories>

<repository>

<id>central</id>

<name>Central Repository</name>

<url>http://repo.maven.apache.org/maven2</url>

<layout>default</layout>

<snapshots>

<enabled>false</enabled>

</snapshots>

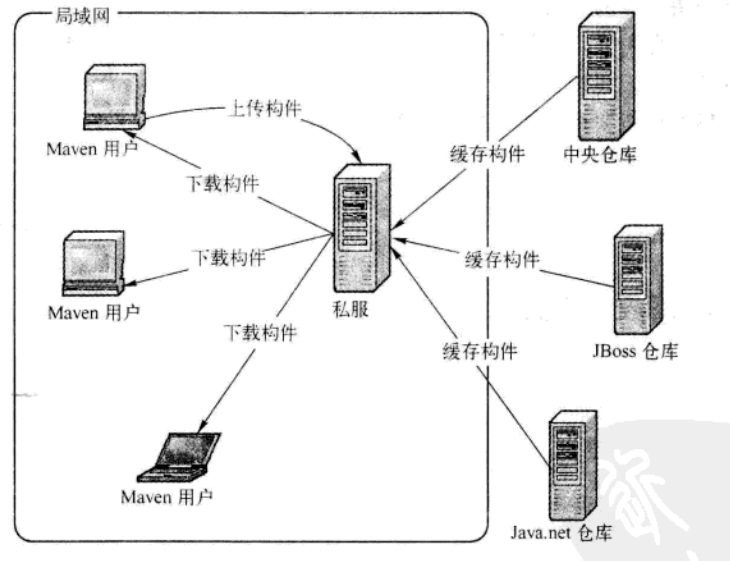
</repository>

</repositories>

中央仓库包含了大多数流行的开源JAVA构件，并包括源码、作者信息、SCM、信息、许可证信息等。

**私服**

私服是一种特殊的远程仓库，它是架设在局域网内的仓库服务，私服代理广域网上的远程仓库，供局域网的Maven用户使用。如图4.2所示



4.2 私服的用途

私服的好处：

* 节省自己的外网带宽
* 加速Maven的构建
* 部署第三方构件。当某个构件无法从远程仓库获得时，可通过建立私服，部署到内部仓库中，供内部的Maven项目使用（如Oracle驱动jar包因版权问题不能放置在开源的远程仓库中）
* 提供稳定性，增强控制。

**远程仓库的设置**

很多情况下，默认的远程仓库无法满足 项目的需求，这时就需要项目构建在另一个远程仓库上，如Jboss仓库。可通过在POM中配置该仓库。

**部署远程仓库**

私服最大作用就是部署第三方构件，包括外部远程仓库无法获得（如涉及版权）、组织内部生成的构件。部署到私服仓库中，供其他团队成员使用。

首先，需要编辑项目的pom.xml文件。配置distributionManagement元素。

待续…