Atitit 依赖管理的艺术 艾提拉著

目录

[1.1. Maven和Gradle对比三大构建工具：Ant、Maven和Gradle。 1](#_Toc1251)

[1.2. 依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制 2](#_Toc11212)

[1.3. Maven的主要功能主要分为5点，分别是依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制。 2](#_Toc8016)

[1.4. 可以用groupId、artifactId、version组成的Coordination（坐标）唯一标识一个依赖 2](#_Toc27130)

[1.5. 有6种scope，分别是complie(默认)、provided、runtime、test、system、import 2](#_Toc16528)

[1.6. Maven特意设置了标准的项目构建周期，其默认的构建周期如下所示： 2](#_Toc26239)

## 历史为什么需要构建根据

来构建项目，只需要简单的执行命令就可以。 对于个人开发有时候我们从开发到测试到最后打包可能都是一个人做，并且用一个开发工具就解决这些了，因为大多数开发工具[IDE](https://www.baidu.com/s?wd=IDE&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)是自带打包功能的。 但是在实际的公司工作中，开发人员与测试人员、部署人员是分开的，部署人员是不会向开发人员要[源代码](https://www.baidu.com/s?wd=%E6%BA%90%E4%BB%A3%E7%A0%81&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)，他们也不会打包，那么怎么部署呢？ 开发人员开发后，确认没问题，将[源代码](https://www.baidu.com/s?wd=%E6%BA%90%E4%BB%A3%E7%A0%81&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)放到版本控制服务器中，并且写一个脚本，这个脚本运行就可以自动打包，然后部署人员运行这个文件打包后部署。 那么这个脚本里面需要关系到目录与加载的文件等，这时候如果每个项目用的文件都不同，目录名字千奇百怪，那么就不容易统一管理，于是构建工具出现了，它规定你的目录必须要如何定制，这样方便统一管理

## **[Maven和Gradle对比](https://www.cnblogs.com/lykbk/p/erwerwerwerwerwerwe.html)**三大构建工具：Ant、Maven和Gradle。

Java世界中主要有三大构建工具：Ant、Maven和Gradle。经过几年的发展，Ant几乎销声匿迹、Maven也日薄西山，而Gradle的发展则如日中天。笔者有幸见证了Maven的没落和Gradle的兴起。Maven的主要功能主要分为5点，分别是依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制。我们可以从这五个方面来分析一下Gradle比起Maven的先进之处。

## 依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制

## Maven的主要功能主要分为5点，分别是依赖管理系统、多模块构建、一致的项目结构、一致的构建模型和插件机制。

## 可以用groupId、artifactId、version组成的Coordination（坐标）唯一标识一个依赖

## 有6种scope，分别是complie(默认)、provided、runtime、test、system、import

## Maven特意设置了标准的项目构建周期，其默认的构建周期如下所示：

<phases>

<phase>validate</phase>

<phase>initialize</phase>

<phase>generate-sources</phase>

<phase>process-sources</phase>

<phase>generate-resources</phase>

<phase>process-resources</phase>

<phase>compile</phase>

<phase>process-classes</phase>

<phase>generate-test-sources</phase>

<phase>process-test-sources</phase>

<phase>generate-test-resources</phase>

<phase>process-test-resources</phase>

<phase>test-compile</phase>

<phase>process-test-classes</phase>

<phase>test</phase>

<phase>prepare-package</phase>

<phase>package</phase>

<phase>pre-integration-test</phase>

<phase>integration-test</phase>

<phase>post-integration-test</phase>

<phase>verify</phase>

<phase>install</phase>

<phase>deploy</phase>

Maven和Gradle对比 - 微笑点燃希望 - 博客园.html