# 安装目录分析

Maven的安装目录有：

--bin

--boot

--conf

--lib

--LICENSE.txt

--NOTICE.txt

--README.txt

## bin：

该目录包含了mvn运行的脚本，这些脚本用来配置Java命令，准备好classpath和相关的Java系统属性，然后执行Java命令。其中mvn是基于UNIX平台的shell脚本，mvn.bat是基于windows平台的bat脚本。在命令行输入任何一条mvn命令时，实际上就是在调用这些脚本。

该目录还包含了mvnDebug和mvnDebug.bat两个文件，同样，前者是UNIX平台的shell脚本，后者是Windows平台的bat脚本。

那么mvn和mvnDebug有什么区别和联系呢？打开文件我们就可以看到，两者基本是一样的，只是mvnDebug多了一条MAVEN\_DEBUG\_OPTS配置，其作用就是在运行Maven时开启debug，以便调试Maven本身。此外，该目录还包含m2.conf文件，这是classword的配置文件，后面会介绍classword。

## boot：

该目录只包含一个文件，以Maven3.0为例，该文件为plexus-classwords-2.2.3.jar。plexus-classwords是一个类加载器框架，相对于默认的java类加载器，它提供了更丰富的语法以方便配置，Maven使用该框架加载自己的类库。更多关于classwords的信息请参考http://classwords.codehuas.org。对于一般的Maven用户来说，不必关心该文件。

## conf：

该目录包含了一个非常重要的文件settings.xml。直接修改该文件，就能在机器上全局的定制Maven的行为。一般情况下，我们更倾向于复制该文件至~/.m2/目录下，然后修改该文件，在用户范围定制Maven的行为。后面将会多次提到settings.xml，并逐步分析其中的各个元素。

## lib：

该目录包含了所有Maven运行时需要的Java类库，Maven本身是分模块开发的，因此用户能看到诸如maven-core-3.0.jar、maven-model-3.0.jar之内的文件。此外，这里还包含一些Maven用到的第三方依赖，如commons-cli-1.2.jar、google-collection-1.0.jar等。对于Maven2来说，该目录只包含一个如maven-2.2.1-uber.jar的文件，原本各为独立Jar文件的Maven模块和第三方类库都被拆解后重新合并到了这个JAR文件中。可以说，lib目录就是真正的Maven。关于该文件，还有一点值得一提的是，用户可以在这个目录中找到Maven内置的超级POM。