

《软件工程》

实验报告四

|  |  |
| --- | --- |
| 班 级： | **计163** |
| 学 号： | **10161785** |
| 姓 名： | **林宇翩** |
| 指导教师： | **阮彤** |

信息科学与工程学院

2019年 4月

**实验四 Build Maven**

1. **Maven是什么？**

Maven项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的项目管理工具软件。它是一个项目管理工具，包含了一个项目对象模型 (Project Object Model)，一组标准集合，一个项目生命周期(Project Lifecycle)，一个依赖管理系统(Dependency Management System)，和用来运行定义在生命周期阶段(phase)中插件(plugin)目标(goal)的逻辑。当你使用Maven的时候，你用一个明确定义的项目对象模型来描述你的项目，然后Maven可以应用横切的逻辑，这些逻辑来自一组共享的（或者自定义的）插件。Maven具有两个主要的特征：具有相同的项目结构（pom.xml、src/main/java、src/test/java）和统一维护jar包。

1. **Maven的常用命令**

mvn archetype:generate 创建Maven项目

mvn compile 编译源代码

mvn deploy 发布项目

mvn test-compile 编译测试源代码

mvn test 运行应用程序中的单元测试

mvn site 生成项目相关信息的网站

mvn clean 清除项目目录中的生成结果

mvn package 根据项目生成的jar

mvn install 在本地Repository中安装jar

mvn eclipse:eclipse 生成eclipse项目文件

mvnjetty:run 启动jetty服务

mvntomcat:run 启动tomcat服务

mvn clean package -Dmaven.test.skip=true:清除以前的包后重新打包，跳过测试类

1. **Build Maven 实践**

**1.检查MVA.bat的位置：**

C:\maven\apache-maven-3.0.5\bin

**查看Path:**

C:\Program Files\AMD APP\bin\x86;

%JAVA\_HOME%\bin;

%CATALINA\_HOME%\bin;

C:\Program Files\Intel\iCLS Client\;

%SystemRoot%\system32;

%SystemRoot%;

%SystemRoot%\System32\Wbem;

%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;

C:\Program Files\Intel\OpenCL SDK\2.0\bin\x86;

C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;

C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;

C:\Program Files\Microsoft SQL Server\100\Tools\Binn\;

C:\Program Files\Microsoft SQL Server\100\DTS\Binn\;

C:\Program Files\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\;

C:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\PrivateAssemblies\;

C:\Program Files\Rational\common;C:\Program Files\Autodesk\Backburner\;

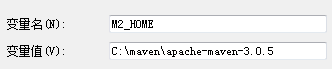
C:\Program Files\Common Files\Autodesk Shared\;

C:\Program Files\ATI Technologies\ATI.ACE\Core-Static;

**%M2\_HOME%\bin;**

C:\Program Files\TortoiseSVN\bin

**查看M2\_HOME:**



**2. 检察待编译文件的目录结构，包括**

1. **pom.xml文件所在目录**
2. **src文件目录**
3. **.java源文件所在目录**
4. **Junit测试文件所在目录**

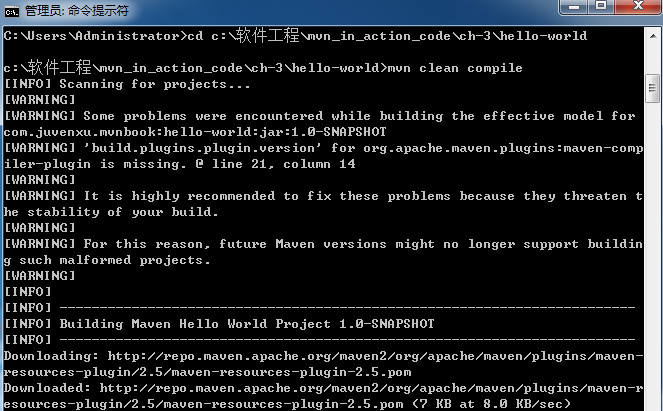
**3. 在系统目录下键入mvn clean compile,结果如下:**

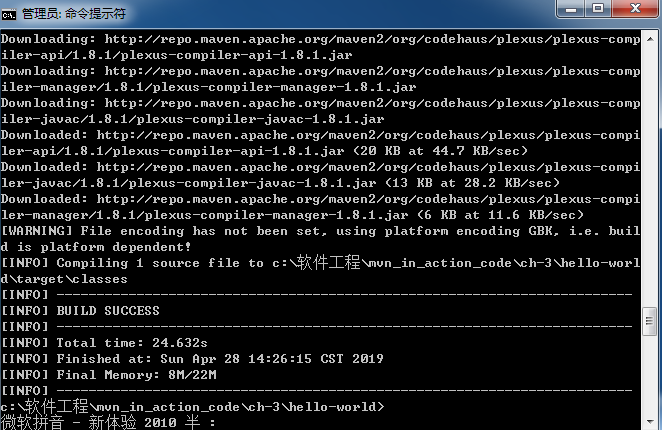


**运行之后发现BUILD FAILURE,因为路径不对，找不到pom.xml**

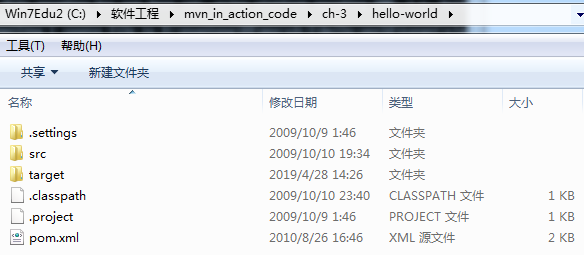
**4.查找之后进入pom.xml的目录C:\软件工程\mvn\_in\_action\_code\ch-3\hello-world：**

**再次键入mvn clean compile,查看编译结果BUILD SUCCESS:**

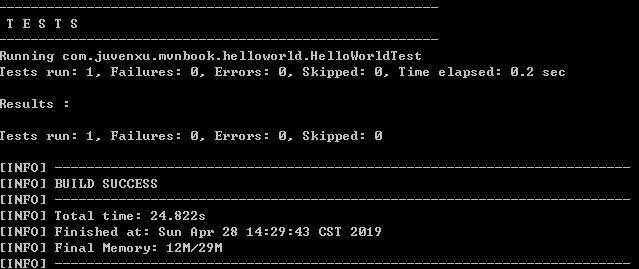




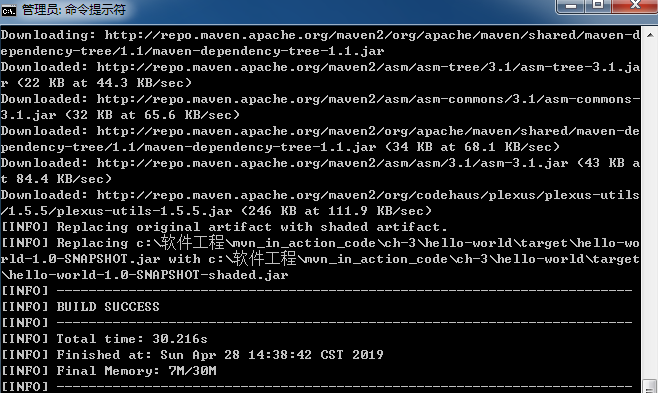
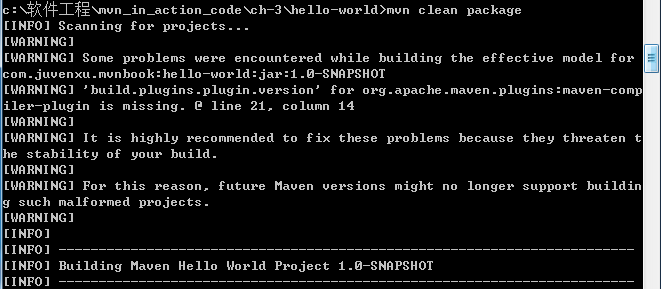
**结果新增target目录：**



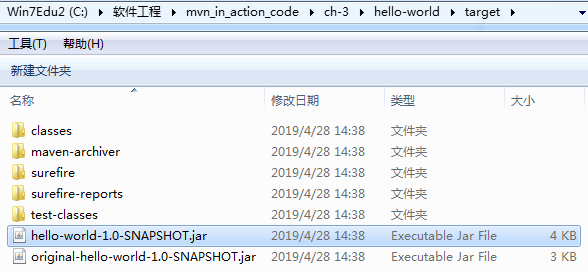
**5.键入命令mvn clean test测试一下:**



**6.键入mvn clean package:**

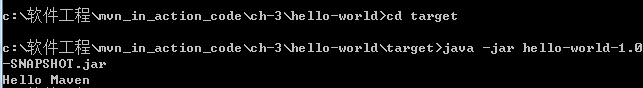


**查看target目录，存在hello-world-1.0-SNAPSHOT.jar文件：**



**7.Cd命令进入target目录下，键入java –jar hello-world-1.0-SNAPSHOT.jar**

**运行hello-world-1.0-SNAPSHOT.jar文件，结果如下：**



**输出结果：Hello Maven**

1. **收获与总结**

通过本次实验，了解了Maven的特征与作用，初步熟悉了其常用的命令，并在此基础上对Maven进行了Build实践，加深对命令的使用熟练程度，并对其导致的文件结构的变化有了基础性的了解。