



Lab 0: Java+Eclipse+Maven 开发环境配置



开发环境

- **JDK 8 (for Windows, Linux, or OS X)**
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- **Eclipse IDE** <http://www.eclipse.org/downloads>
 - Eclipse IDE for Java Developers (Lab1、Lab4、Lab6、Lab7使用)
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/neonr>
 - Eclipse IDE for Java EE Developers (Lab2使用)
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/neonr>
- **Maven** <http://maven.apache.org> (可选)
 - 某些Eclipse版本中已经集成了Maven

Java



- 组成部分：
 - JDK: Java Development Kit 开发者工具包
 - JRE: Java Runtime Environment 运行环境
 - JVM: Java Virtual Machine 虚拟机，对Java代码解释执行
- 多种版本的JDK/JRE: Java SE, Java EE, Java ME, Java Mobile, etc
- Java环境变量：
 - JAVA_HOME指向JDK安装目录；
 - PATH中建立指向JDK和JRE的bin目录的path变量；
 - CLASSPATH中建立指向JAVA基础类库包的变量。

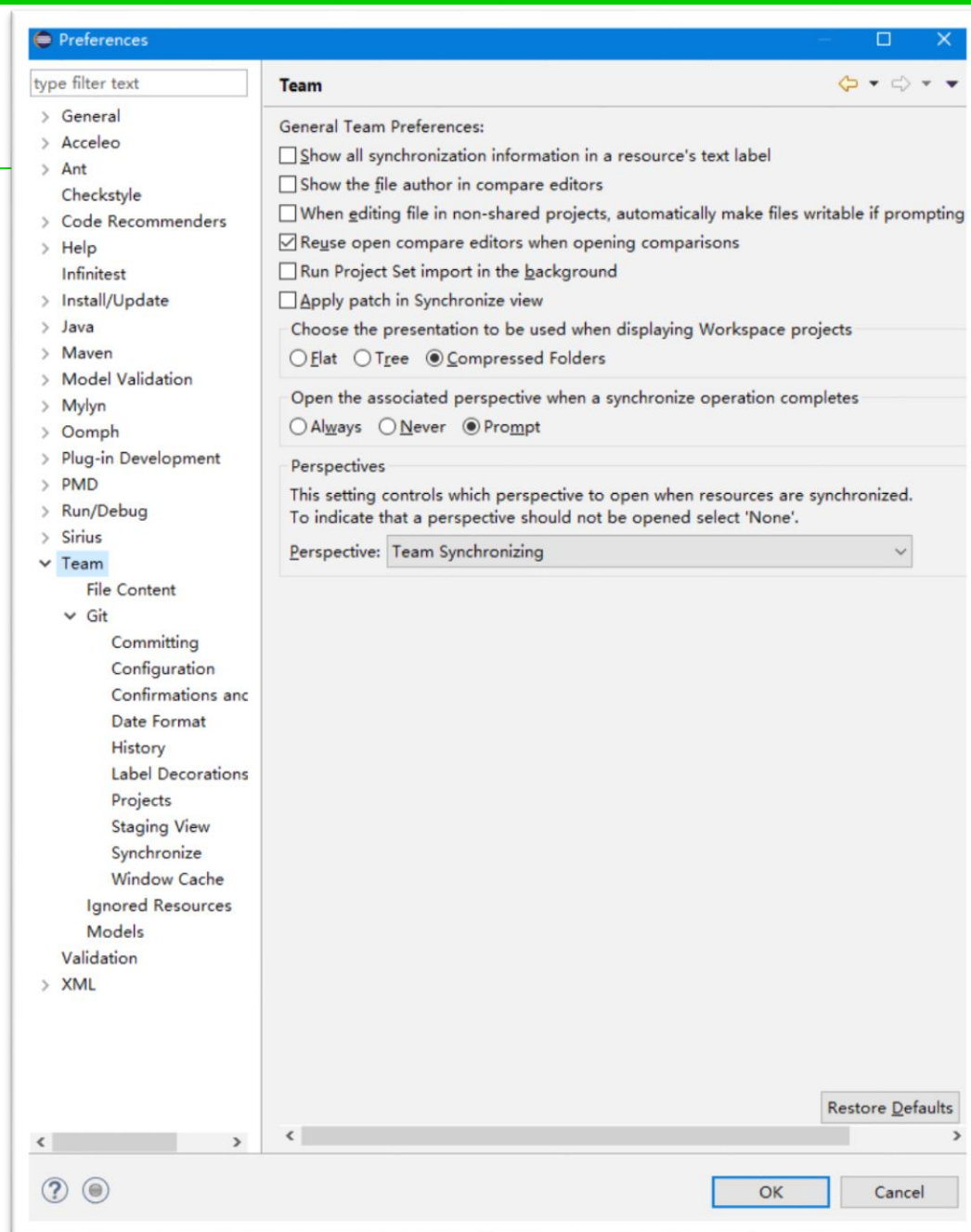
Eclipse

- Java的集成开发环境IDE
- 最初由IBM主导开发，后来变为开源项目，是目前最流行的Java IDE。
- 本身是一个框架平台，第三方提供丰富的Java开发插件，使得Eclipse具有丰富的扩展性和灵活性。
- 可通过Eclipse Marketplace方便的查询和安装各种plugin。
- 其他Java IDE: NetBeans, JBuilder, JDeveloper, etc



配置Eclipse

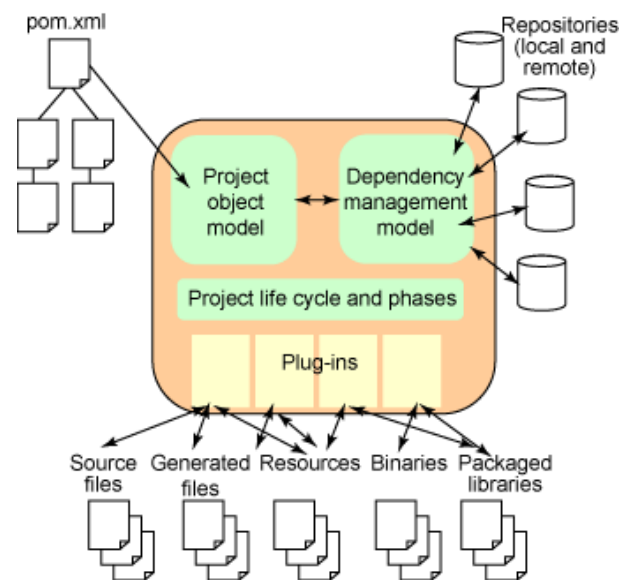
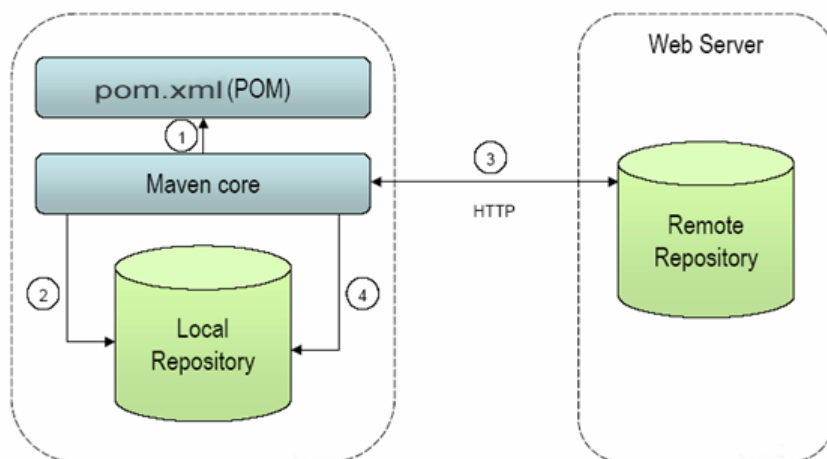
- Eclipse的workspace目录：工作区，开发的所有项目均放置在此处，每个project一个文件夹；
- 可以通过配置更改项目的存储位置，也可以通过import将磁盘上任何位置的project加入到Eclipse中；
- 通过Window->Preferences打开配置窗口，更改任何相关配置信息。



Maven



- 来自Apache的开源项目，支持Java项目的标准化自动构建 (build)。
 - 项目对象模型POM: pom.xml
 - 依赖项管理模型: 自动管理
 - 构建生命周期和阶段: 对项目构建的各阶段任务，分别使用plug-in加以支持。

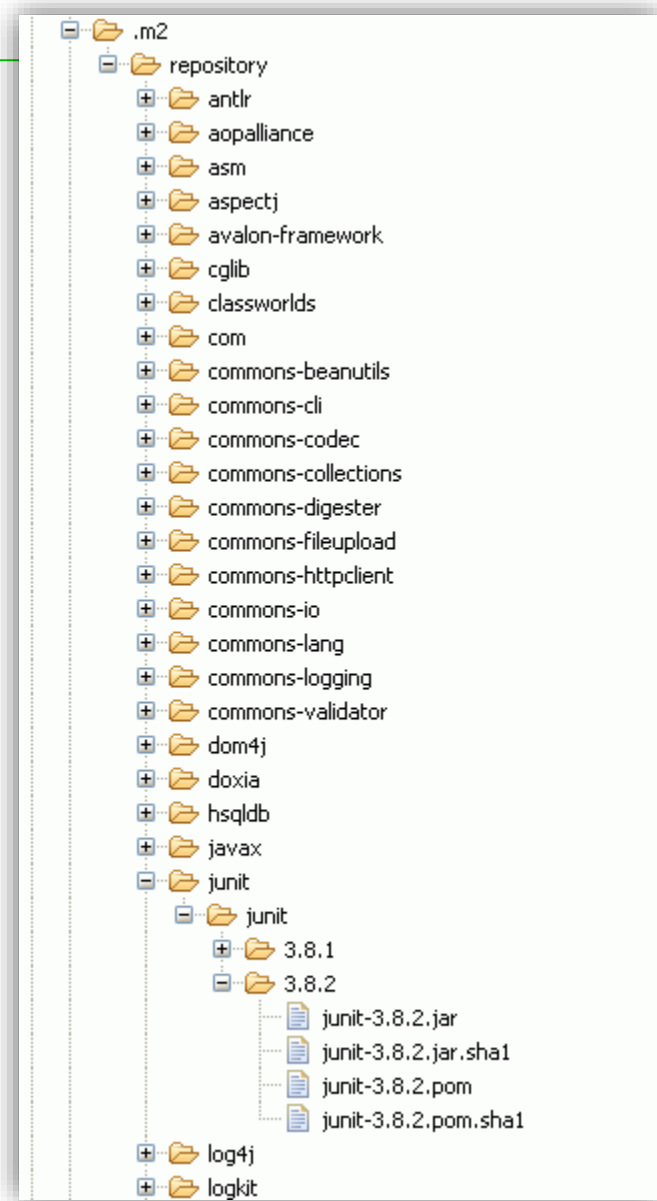


Maven POM

- 只需在项目的pom.xml简单声明，Maven可自动将所需依赖项从远程服务器下载到本地并加入项目，省去了繁琐的手工配置过程：

```
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
  </dependency>
</dependencies>
```

- 这些下载的JAR在本地由Maven统一管理，在各个本地项目之间共享。

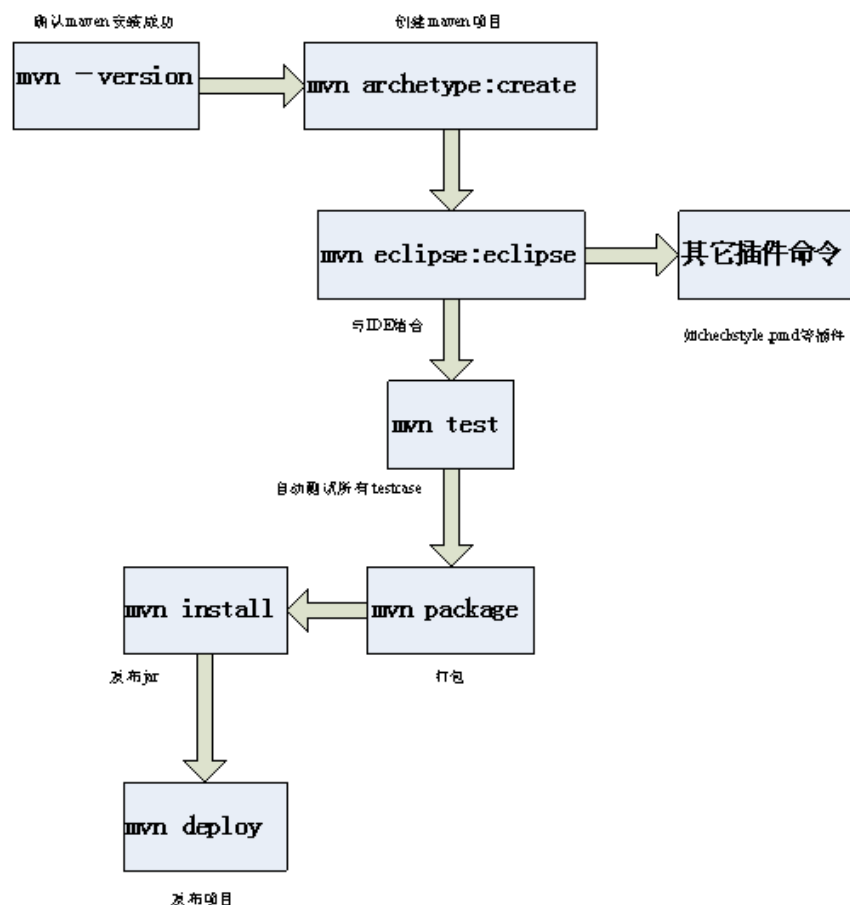


Maven的project目录结构

- **Maven**提供了丰富的ArcheType，可根据模板创建标准的Maven项目结构，从而使开发人员更容易理解项目，为以后的维护工作打下良好的基础；
- **Maven**根据缺省项目目录，预定义了相关的build动作，无需人工配置和干预。

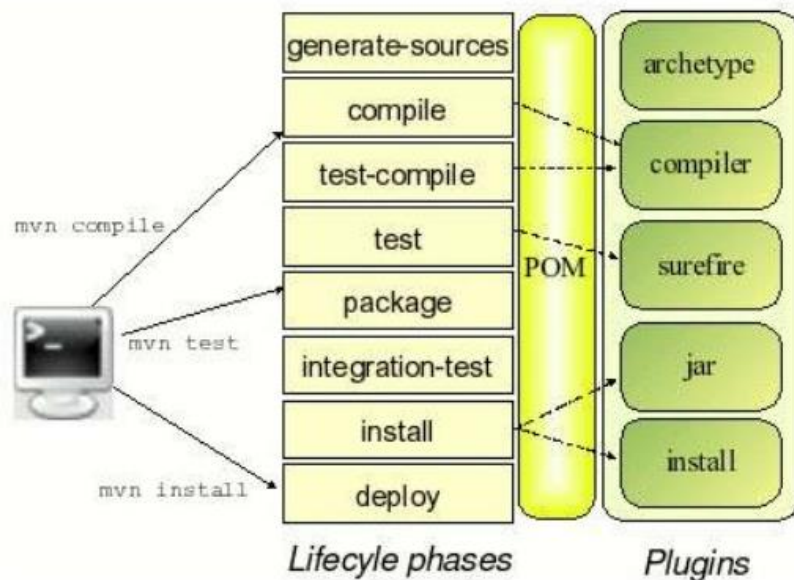
项目根/	
pom.xml	Maven2 的 pom.xml 文件
src/	
main/	项目主体目录根
java	源代码目录
resources	所需资源目录
filters	资源过滤文件目录
assembly	Assembly descriptors
config	配置文件目录根
test/	项目测试目录根
java	测试代码目录
resources	测试所需资源目录
filters	测试资源过滤文件目录
site	与 site 相关的资源目录
target/	输出目录根
classes	项目主体输出目录
test-classes	项目测试输出目录
site	项目 site 输出目录

Maven Build生命周期阶段



- **Validate:** 验证工程是否正确，所有需要的资源是否可用。
- **Compile:** 编译项目的源代码。
- **Test-compile:** 编译项目测试代码。
- **Test:** 使用已编译的测试代码，测试已编译的源代码。
- **Package:** 将已编译的源代码打包，如jar。
- **Integration-test:** 在集成测试可以运行的环境中处理和发布包。
- **Install:** 把包安装在本地的repository中，可以被其他工程作为依赖来使用。
- **Deploy:** 在整合或者发布环境下执行，将最终版本的包拷贝到远程的repository，使得其他的开发者或者工程可以共享。
- **Clean:** 清除之前产生的build结果。

Maven Build生命周期阶段



- 每个Maven阶段命令通过调用Maven插件来运行；
- 每个被调用的阶段命令自动调用其生命周期前面的阶段命令
 - 例如mvn package 会依次运行 compile , test 阶段命令。

Maven的运行方式

- 一般情况下，在命令行执行maven指令 (clean, compile, test, package, install, deploy, ...).
- 在Eclipse中可以以plugin的形式加入maven支持，直接创建符合Maven标准的project，并加以build;
- 对Eclipse创建和管理的project，需要手工调整目录结构，使之符合Maven标准。

The screenshot displays the Eclipse IDE interface. The top console shows the Maven build process for a project named 'HelloWorld'. The build is successful, with the following output:

```
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
PS C:\Users\win64> cd C:\tmp\maven\myapp\HelloWorld
PS C:\tmp\maven\myapp\HelloWorld> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building HelloWorld 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.5:resources (default-resources) @ HelloWorld ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\tmp\maven\myapp\HelloWorld\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:compile (default-compile) @ HelloWorld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.5:testResources (default-testResources) @ HelloWorld ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\tmp\maven\myapp\HelloWorld\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ HelloWorld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ HelloWorld ---
[INFO] Surefire report directory: C:\tmp\maven\myapp\HelloWorld\target\surefire-reports
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running com.pstreets.mavendemo.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.04 sec
[INFO]
[INFO] Results :
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] --- maven-jar-plugin:2.3.2:jar (default-jar) @ HelloWorld ---
[INFO] Building jar: C:\tmp\maven\myapp\HelloWorld\target\HelloWorld-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 7.760s
[INFO] Finished at: Sun Sep 16 21:40:19 UST 2012
[INFO] Final Memory: 6M/15M
[INFO]
[INFO] -----
PS C:\tmp\maven\myapp\HelloWorld>
```

The Package Explorer on the left shows the project structure. The Run As context menu is open, showing the following options:

- Run As
- Debug As
- Profile As
- Coverage As
- Restore from Local History...
- Acceleo
- Checkstyle
- Maven
- PMD
- Team
- Compare With
- Configure
- Source
- Migrate to UML Designer 4.0
- Properties

The Run As context menu is also open, showing the following options:

- m2 1 Maven build (Alt+Shift+X, M)
- m2 2 Maven build...
- m2 3 Maven clean
- m2 4 Maven generate-sources
- m2 5 Maven install
- m2 6 Maven test
- Run Configurations...

The bottom status bar shows the progress of the build, indicating that no operations are to be displayed at this time.



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

Enjoy Java development
in this semester!

