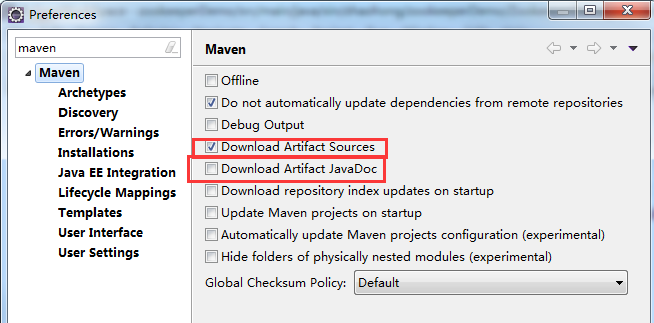
Eclipse中Maven下载源代码

# 在Eclipse中利用Maven下载依赖的源代码、文档等

**Preferences**🡺搜索maven==> 选择**Download Artifact Sources;就会自动下载源码了。**



# 下载的依赖版本不兼容问题的解决

问题描述：

当使用Maven添加一个依赖时，所添加的依赖同时也依赖很多其他依赖，而当前工程恰好有依赖了这些其他依赖中的某些，此时两种依赖的版本就有可能发生冲突，报出异常，此时可以通过**排除依赖的某个依赖，解决冲突**。

具体来讲：

举例：在最近学习Dubbo时，Dubbo中使用了**Spring 3.4.2版本**，而当前工程使用的是**Spring 4.4.2版本**，这时候运行当前版本就会出错。

**解决方法：在引入Dubbo的时候，排除Spring 3.4.2这个版本。**

**排除exclusions**：



# 打jar包与war包

## 打包方法

无论jar包还是war包都是：

点击项目名称，右击选择Export，然后选择jar或war即可。

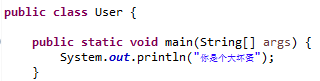
无论jar包还是war包，名字都是随便起的，不影响内部类等的使用。

## 打出jar包



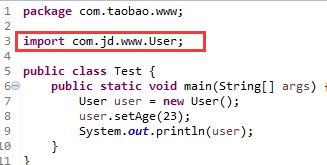
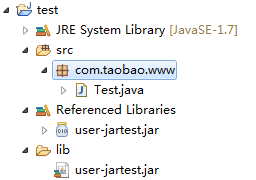
## Runnable JAR file可执行的jar包

在命令行下，利用java –jar xxx.jar运行即可。



## JAR file：普通的jar包

一般见到的都是这种jar包。可以调用其内部的类。



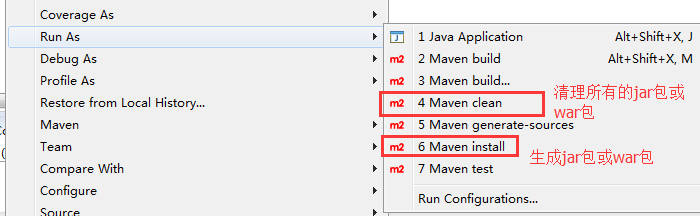
注意：jar包的名字可以随意起，不一定和工程名一样。

## 普通web工程打出war包

到处的war包，放到tomcat的**webapps**目录下，即可运行。

## Maven工程打包

点击工程名字，右击，执行**Run As**，**Maven install**。





Maven 工程的**maven install** 会把jar包或war包存储在**target**目录下，同时拷贝至**本地的仓库**中。

## Maven-web与Java-Web工程之间的转化

转化主要是依赖的转化。

### Java-web转为Maven-web

只需要把**所有的jar包找到对应的groupId和artifactId以及版本号**，在pom.xml文件中添加依赖就可以。

### Maven-web转为java-web

点击项目名称，右击选择Run As，**选择maven-install，然后maven工程会将pom中的所有依赖，生成一个war包，这个war包的lib文件夹中存储了所有的jar**包。因此解压war包，复制lib文件夹导入Java-web工程的lib文件夹下即可。