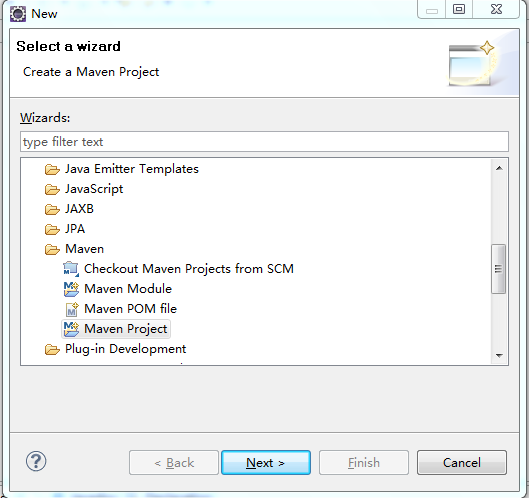
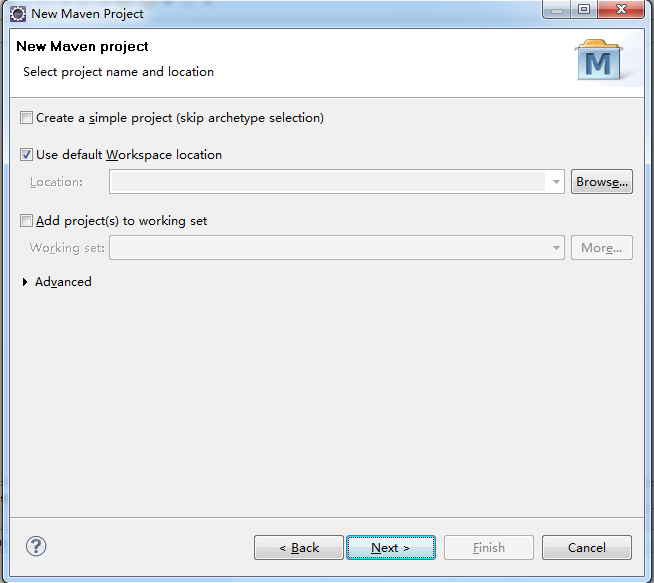
使用Eclipse的maven构建一个web项目

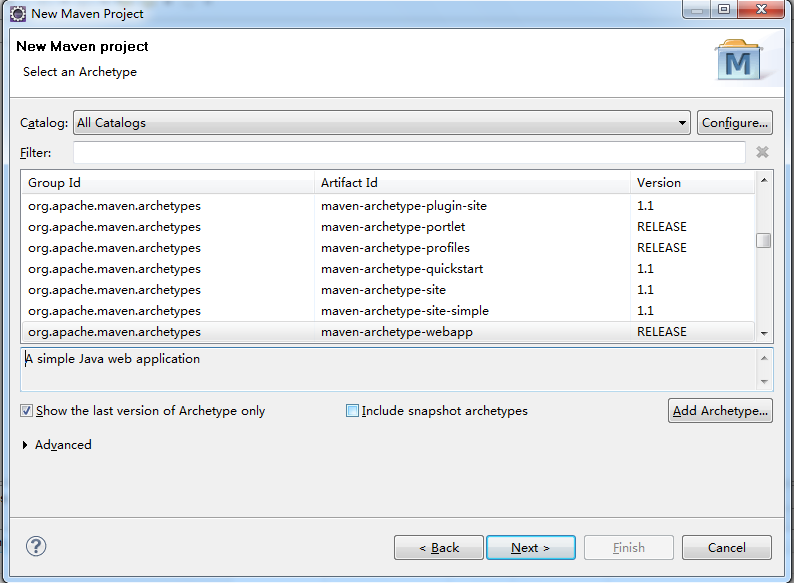
1、选择建立Maven Project 选择File -> New -> Other，在New窗口中选择 Maven -> Maven Project；点击next



2、选择项目路径 Usedefault Workspace location默认工作空间



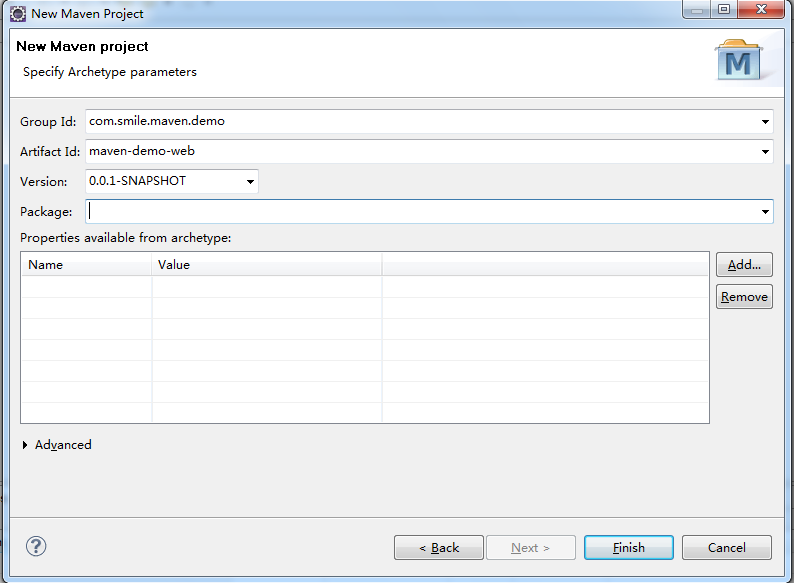
3、选择项目类型 在Artifact Id中选择maven-archetype-webapp



4 、输入Group ID和 Artifact ID，以及Package

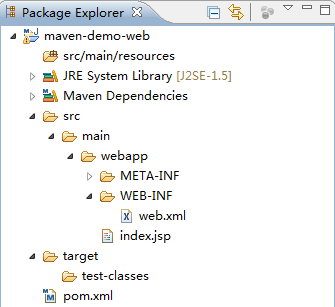
Group ID一般写大项目名称。Artifact ID是子项目名称。

Package是默认给你建一个包，不写也可以



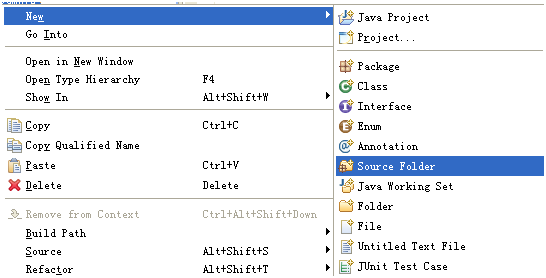
5、刚建立好后的文件结构如下图

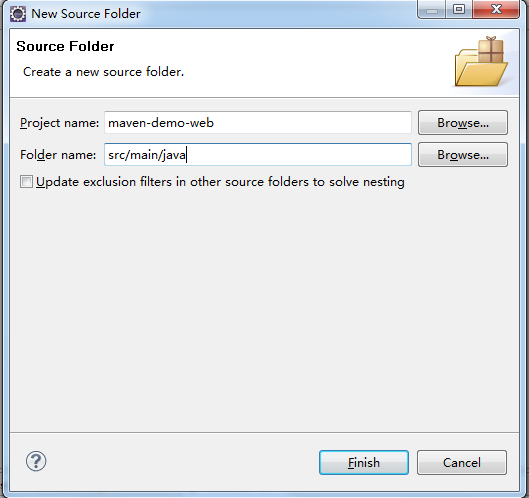
如果这里显示的内容多，一般是Filters设置的问题；或perspective为JavaEE模式，改成Java模 式就可以了



6、配置项目 需要添加src/main/java，src/test/java ，src/test/resources三个文件夹。右键项目根目录点击New -> Source Folder，建出这三个文件夹

注意：不是建普通的Folder，而是Source Folder





7、更改class路径 右键项目，Java Build Path -> Source 下面应该有4个文件夹。src/main/java，src/main /resources，src/test/java ，src/test/resources

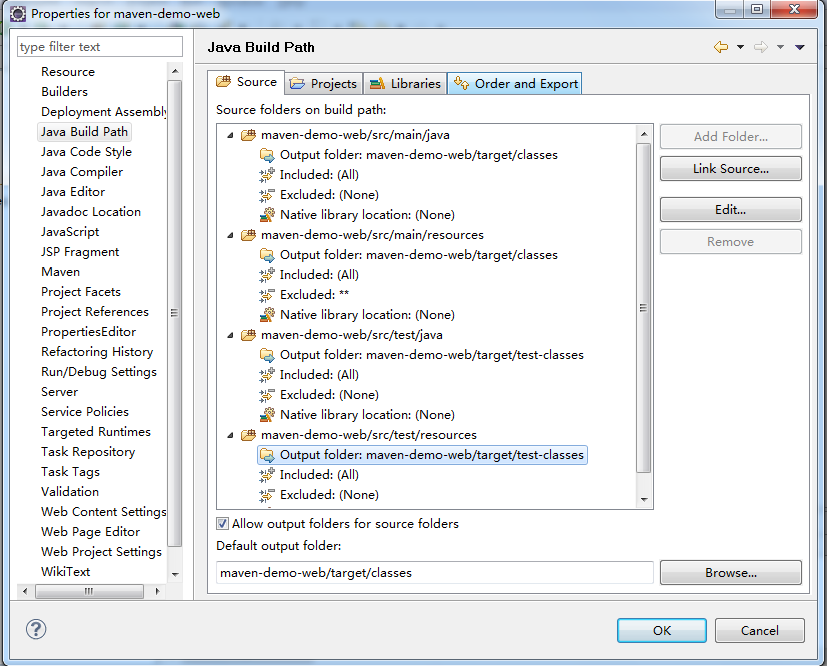
选上Allow output folders for source folders

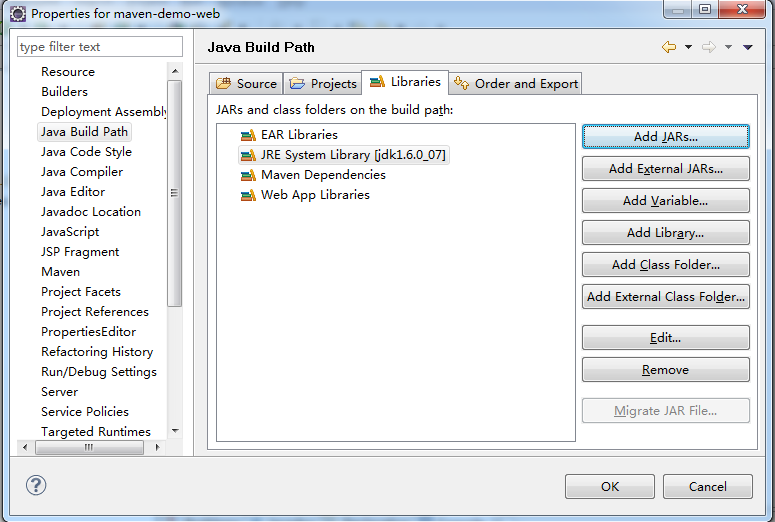
双击每个文件夹的Output folder，选择路径

src/main/java，src/main/resources，选择target/classes;

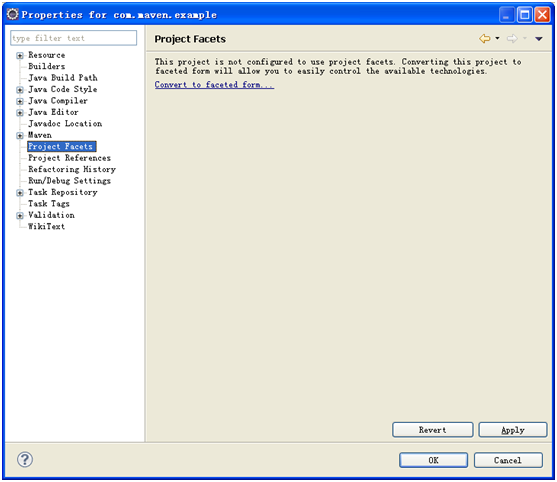
src/test/java ，src/test/resources, 选择target/test-classes;

在此处还要更改：更改文件夹显示的顺序：点击Order and Export；更改JDK版本：在Libraries双击JRE System Library，要1.6版本

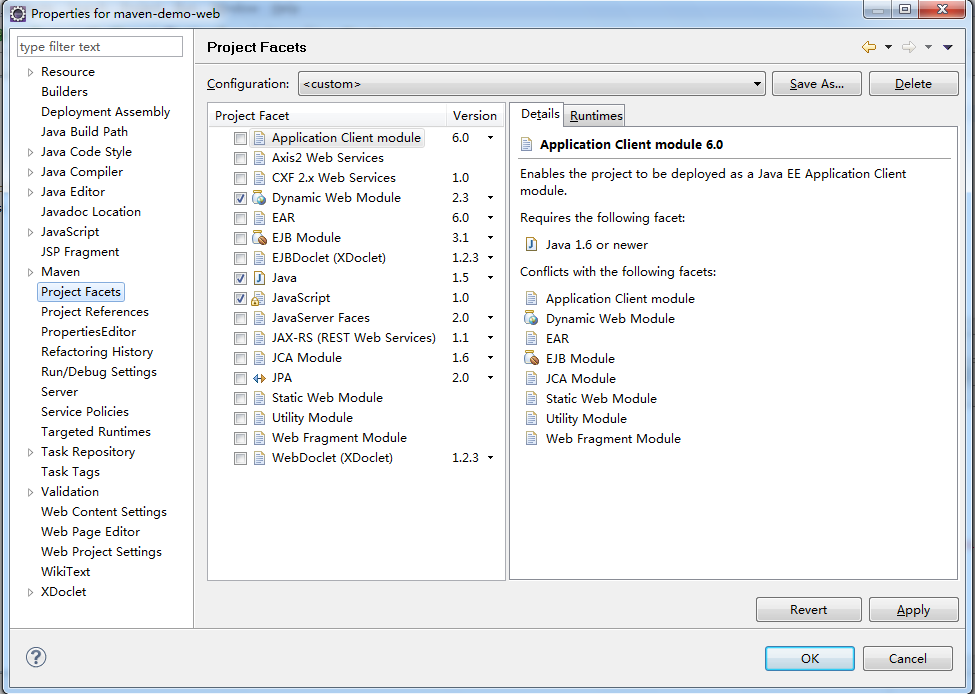




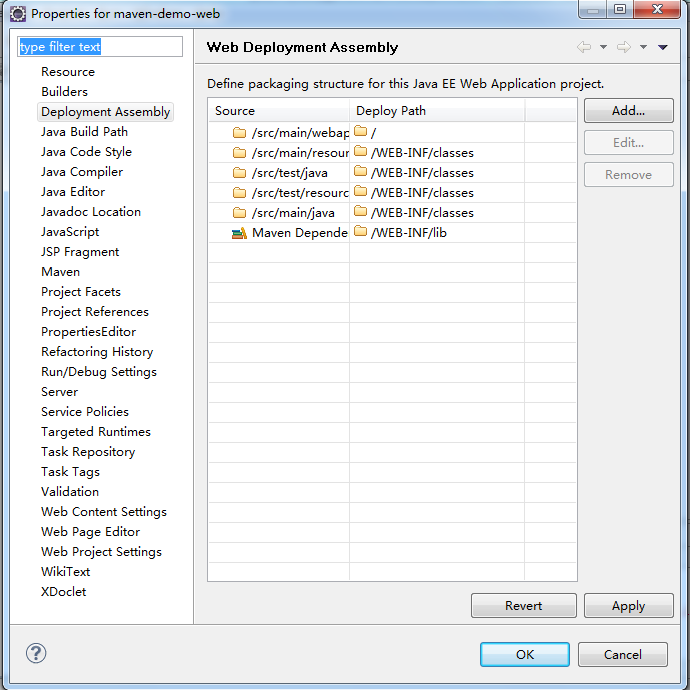
8、把项目变成Dynamic Web项目 右键项目，选择Project Facets，点击Convert to faceted fro



9、配置Project Facets 更改Dynamic Web Module的Version为2.3。(3.0为Java7的)。  
如果提示错误，可能需要在Java Compiler设置Compiler compliance level 为1.6，或者需要在此窗口的Java的Version改成1.6



10、设置部署程序集(Web Deployment Assembly)  
上面步骤设置完成后，点击OK，Properties窗口会关 闭，在右键项目打开此窗口。在左侧列表中会出现一个Deployment Assembly，点击进去后，如下图



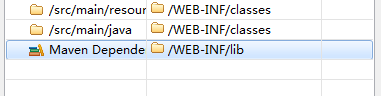
此处列表是，部署项目时，文件发布的路径。

（1）我们删除test的两项，因为test是测试使用，并不需要部署。

（2）设置将Maven的jar包发布到lib下。

Add -> JavaBuild Path Entries -> Maven Dependencies -> Finish

设置完成效果图



11、构建框架 在pom.xml中添加所需要的jar包

使用Maven POM editor打开项目中的pom.xml文件，选择Dependencies，在Dependencies栏目点击Add进行，首先弹出一个搜索按钮，例 如输入jsf,就会自动搜索关于JSF相关的jar包，我们选择2.0.4版本的jsf，将jsf包全部添加进来

需要添加的其他jar包有：junit、jstl

或者点击pom.xml直接编辑pom.xml文件，这样可以直接copy过来dependencies内容；

我们的pom.xml文件可直接复制下面的内容：

<project xmlns="http://maven.apache.org/POM/4.0.0"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>  
<groupId>com.smile.maven.demo</groupId>  
<artifactId>maven-demo-web</artifactId>  
<packaging>war</packaging>  
<version>0.0.1-SNAPSHOT</version>  
<name>maven-demo-web Maven Webapp</name>  
<url>http://maven.apache.org</url>  
  
<dependencies>  
<dependency>  
<groupId>junit</groupId>  
<artifactId>junit</artifactId>  
<version>3.8.1</version>  
<scope>test</scope>  
</dependency>  
<dependency>  
<groupId>com.sun.faces</groupId>  
<artifactId>jsf-api</artifactId>  
<version>2.0.4-b09</version>  
</dependency>  
<dependency>  
<groupId>com.sun.faces</groupId>  
<artifactId>jsf-impl</artifactId>  
version>2.0.4-b09</version>  
</dependency>  
<dependency>  
<groupId>javax.servlet</groupId>  
<artifactId>jstl</artifactId>  
<version>1.2</version>  
</dependency>  
<dependency>  
<groupId>taglibs</groupId>  
<artifactId>standard</artifactId>  
<version>1.1.2</version>  
</dependency>  
</dependencies>  
  
<build>  
<finalName>maven-demo-web</finalName>  
</build>  
  
</project>

12、发布 对着工程点右键:Run As ->Maven install 然后再Run As -> Mavenpackage.  
生成完后用Tomcat跑即可