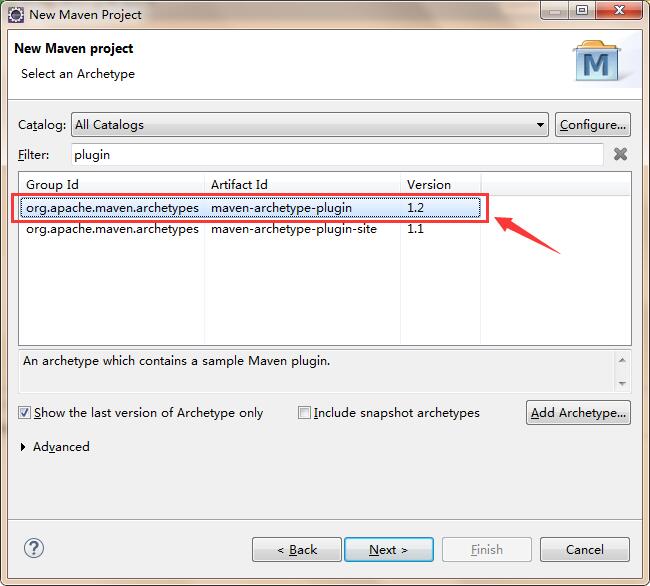
# **[Maven3.x 插件开发入门](http://www.cnblogs.com/dreamroute/p/5262890.html)**

　　Maven工具有很多插件，各种各样的插件，让我们开发调试过程中非常方便，但是终究是有你想要的但是现目前插件不能满足的（可能性非常非常低），这个时候就需要使用其他的替代工具，或者是自己来开发一个Maven插件。

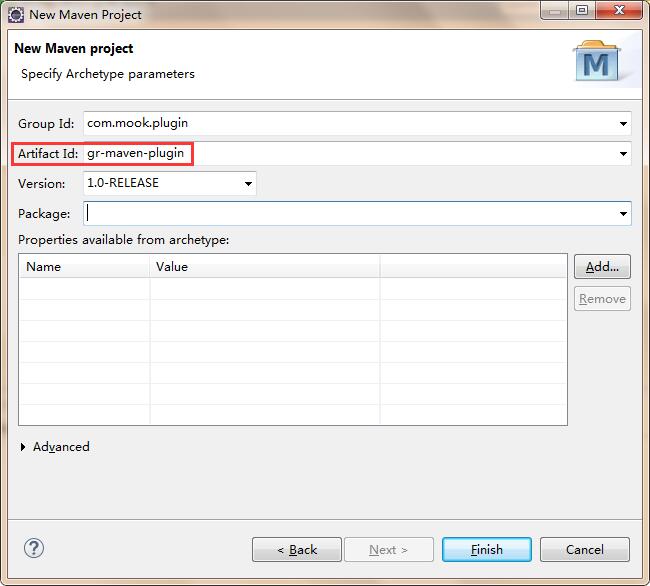
　　事实上，学Maven插件开发对于我们这种小鸟意义并不大，有两个原因：1、我们平常需要使用的，几乎Maven都有提供；2、如果实在没有，那一般来说都有类似的工具作为替代，我们可以去使用另外的工具。但是，我们对Maven使用，研究深入了以后，学学他的插件开发来玩一下还是不错的。这对Maven的精髓思想的理解还是有很大帮助的。

　　这里的的插件开发工具是Eclipse-Mars版本。

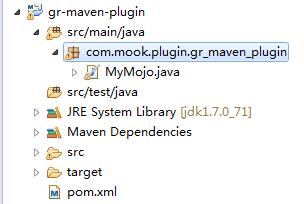
　　1、创建插件项目：（项目原型选择插件类型）



　　2、Artifact Id 以xxx-maven-plugin命名，如：gr-maven-plugin，下面Version的内容无所谓。



　　3、创建完毕之后的结构是这样：



　　4、修改pom.xml文件，加上2个依赖：分别是maven-plugin-api和maven-plugin-annotations，前者是插件开发API，后者是插件中使用的注解定以的包，注意打包方式为：<packaging>maven-plugin</packaging>。完整的pom.xml文件如下，一定要把自动生成那些没用的东西删掉，只留下下面的内容，否则运行插件的时候有可能报错。

按 Ctrl+C 复制代码

按 Ctrl+C 复制代码

　　5、删掉默认的包，自己新建一个包com.mook.plugin.gr,在这个包下面创建一个类叫做Car，继承AbstractMojo类。重写里面的execute方法。如下：

[IMG_259](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

@Mojo(name = "drive")public class Car extends AbstractMojo {

@Override

public void execute() throws MojoExecutionException, MojoFailureException {

System.out.println("Car drive...");

}

}

[IMG_260](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

　　6、这样插件就开发完成了。我们将插件install到本地仓库。然后在项目组引入，可以是在本插件项目中引入，也可以在其他项目中引入。

[IMG_261](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

<build>

<plugins>

<plugin>

<groupId>com.mook.plugin</groupId>

<artifactId>gr-maven-plugin</artifactId>

<version>1.0-RELEASE</version>

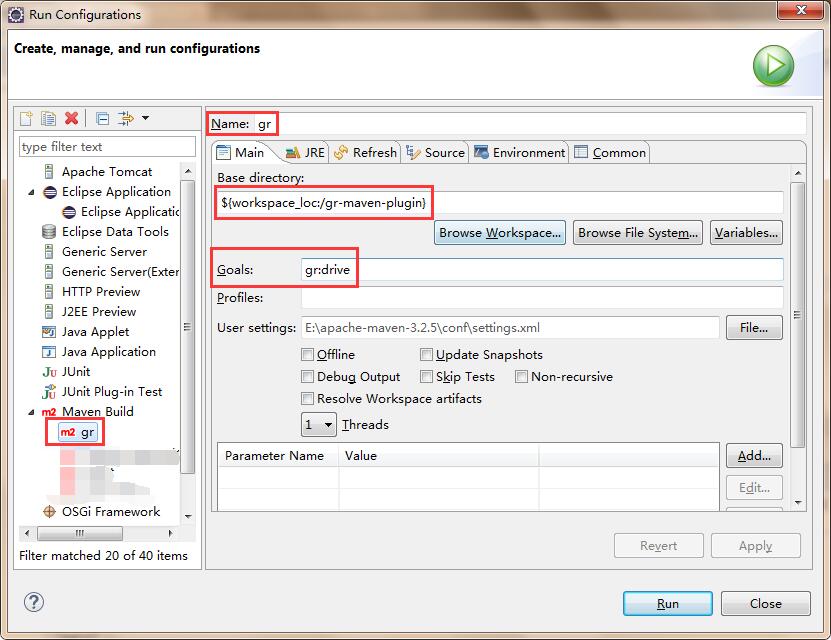
</plugin>

</plugins>

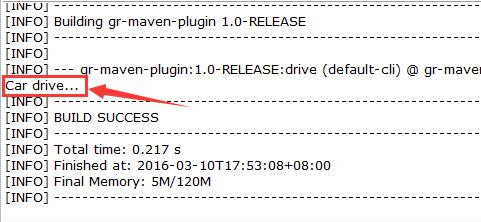
</build>

[IMG_262](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

　　7、使用eclipse的Maven插件来运行，这里的Maven插件实质Eclipse的Maven插件，而不是Maven自己的插件。如下：



　　8、输出结果：Car drive...结果显然是正确的。



　　9、这里来解释下里面有几个细节。

~~1、插件artifactId为什么使用xxx-maven-plugin（或者xxx-plugin-maven），这个是约定的，如果这样命名，在启动插件的时候就可以像上面那样gr:drive。否则就的把gr改成groupId:artifactId:version:xxx这种方式，很不方便，当然也有另外的方式处理，在settings文件里面增加pluginManageMent内容，也能实现简化写法。~~(这个地方改一下，在我们的pom.xml文件中，plugins下面的maven-plugin-plugin中的<goalPrefix>标签下是可以配置这个前缀名称的，比如我这里配置co，那么启动就是co:drive)，如下：

[IMG_265](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-plugin-plugin</artifactId>

<version>3.2</version>

<configuration>

<goalPrefix>co</goalPrefix>

<skipErrorNoDescriptorsFound>true</skipErrorNoDescriptorsFound>

</configuration>

<executions>

<execution>

<id>mojo-descriptor</id>

<goals>

<goal>descriptor</goal>

</goals>

</execution>

<execution>

<id>help-goal</id>

<goals>

<goal>helpmojo</goal>

</goals>

</execution>

</executions>

</plugin>

[IMG_266](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

　　　　2、注解@Mojo是必须要的，这是定义插件对象的启动方法，由于该类只有一个方法，所以启动方法和启动类是一致的。在Maven 3之前是使用注释注解：@goal xxx这种方式。现在已经不使用这种方式了。

　　　　3、我们平时在使用Maven的各种插件的时候往往都能在配置文件中传入属性的值，比如tomcat-maven-plugin插件我们可以随意指定tomcat的端口号。这里插件的处理方式是在Car类中定义一些属性，比如下面这样。然后我们重新将插件install到本地仓库。再次运行。

[IMG_267](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

@Mojo(name = "drive")public class Car extends AbstractMojo {

@Parameter(defaultValue = "8080")

private Integer port;

@Override

public void execute() throws MojoExecutionException, MojoFailureException {

System.out.println("Car drive...");

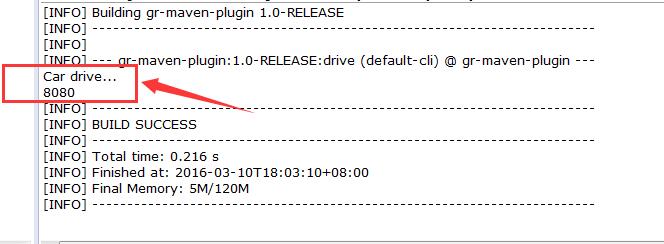
System.out.println(port);

}

}

[IMG_268](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

　　　　输出结果：



　　　　那么，在插件的配置中增加Configuration标签，加上子标签<port>，如下：

[IMG_270](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

<plugin>

<groupId>com.mook.plugin</groupId>

<artifactId>gr-maven-plugin</artifactId>

<version>1.0-RELEASE</version>

<configuration>

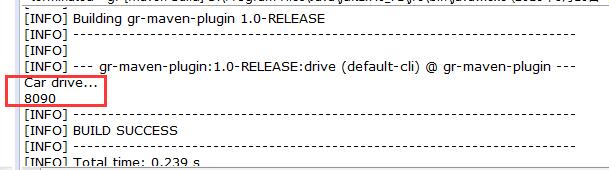
<port>8090</port>

</configuration>

</plugin>

[IMG_271](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

　　　　那么，结果就是8090，这就是插件的参数设置方式。



　　10、到这里基本上就介绍完了。关于插件的运行，关系到Maven的生命周期，阶段和目标这几个概念。这是另外一个话题了。

　　其他：新建的eclipse项目上面报错，显示execution not covered by lifecycle这种错误，搜了一下，在<build>下<pluginManagement>下<plugins>中加入：

<plugin>

<groupId>org.eclipse.m2e</groupId>

<artifactId>lifecycle-mapping</artifactId>

<version>1.0.0</version>

<configuration>

<lifecycleMappingMetadata>

<pluginExecutions>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-plugin-plugin</artifactId>

<versionRange>[3.2,)</versionRange>

<goals>

<goal>descriptor</goal>

<goal>helpmojo</goal>

</goals>

</pluginExecutionFilter>

<action>

<ignore></ignore>

</action>

</pluginExecution>

</pluginExecutions>

</lifecycleMappingMetadata>

</configuration>

</plugin>

[IMG_274](http://www.cnblogs.com/dreamroute/p/javascript:void(0);)

其中：groupId,artifactId,versionRange,goals是需要自己去改变的，因为不有可能其他的插件也会报这个错误，按照这几个标签内容改就行了，我这里是maven-plugin-plugin这个插件在报错。