原文地址：http://blog.csdn.net/natian306/article/details/23247777

### Maven常见问题解决

1. The **[Container](http://lib.csdn.net/base/docker" \o "Docker知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)** 'Maven Dependencies' references non existing library

解决方法，将eclipse中maven插件中“resolve dependencies from workspace projects”的选项取消

默认的本地库更改，修改maven\_home\conf\settings.xml中localRepository的配置

[ERROR] The goal you specified requires a project to execute but there is no POM in this directory (...). Please verify you invoked Maven from the correct directory.

到有pom.xml文件的目录下执行命令

2.      myeclipse从svn导入maven项目，有modules。

myeclipse的check out maven projects from scm，没有试成功怎么导出。但可以先从svn检查maven项目，包含子module，然后再导入exists maven projects。

3. Failure to find xxx:jar:1.0 in ... was cached in the local repository, resolution will not be reattempted until the update interval of nexus has elapsed or updates are forced

      这是一些jar包在之前的仓库无法获得，在更改仓库后，或把jar包部署到了私服上后，仍然出错。这时把.m2 文件夹下对应的 xxx.lastUpdated文件再更新依赖，就可以了。或者使用mvn命令时加 -U参数，忽略xxx.lastUpdated。

      另外以上方式仍然报错，这时可能是maven 库或私服里确实没有对应的jar包，这时可以在私服里增加仓库，或者更改仓库。

4.      pom聚合的install，在maven 的build config里，goals选项里填入install，profiles中填入maven聚合项目中配置的modules profile的id，如果是如下配置：

<profiles>

   <profile>

     <id>modules</id>

     <modules>

    <module>modules/parent</module>

    <module>modules/common</module>

    <module>modules/entity</module>

    <module>modules/ejb</module>

    <module>modules/**[spring](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**</module>

    <module>modules/**[hibernate](http://lib.csdn.net/base/javaee" \o "Java EE知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**</module>

    <module>modules/web</module>

    <module>modules/test</module>

    </modules>

   </profile>

 </profiles>

则这里就要填入modules

5.  maven install的plugin找不到，比如effective pom 中配的是2.3.1，本机上已经有2.2.1了，是不会使用的。就会去maven仓库中下载新版本，maven 的中央库连接经常不稳定。一般会在settings.xml中配置一些镜像。但maven不会依次去尝试所有镜像，而当排在前面的镜像失效、或不能提供最新版本时，就需要调整镜像的位置。

6.  导入的maven项目找不到jdk中的类

jdk版本是1.6，ide是ide出问题了，将Myeclipse重启ide，对有依赖的maven项目进行更新maven依赖、更新maven项目配置操作，然后对本项目clean、 重新build后就可以了。

7.  拷贝依赖包 mvn dependency:copy-dependencies，默认会拷到项目的 target\dependency 目录，想要复制到自定义的目录比如target/libs目录下，需要在pom.xml文件中添加设置覆盖默认设置：

<build>

    <plugins>

        <plugin>

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

            <configuration>

                <outputDirectory>${project.build.directory}/lib</outputDirectory>

                <excludeTransitive>false</excludeTransitive>

                <stripVersion>true</stripVersion>

            </configuration>

        </plugin>

其中${project.build.directory}是maven变量，表示target目录。如果不写的话，将在根目录下创建lib目录。

excludeTransitive，表示是否不包含间接依赖的包；

stripVersion表示复制的jar文件去掉版本信息。

如果需要在其他过程，比如package中加入copy-dependencies，需要在该plugin标签中这样设置：

<executions>

    <execution>

        <id>copy-dependencies</id>

        <phase>package</phase>

        <goals>

            <goal>copy-dependencies</goal>

        </goals>

        <configuration>

            <outputDirectory>libs</outputDirectory>

            <excludeTransitive>false</excludeTransitive>

            <stripVersion>true</stripVersion>

        </configuration>

    </execution>

</executions>

8. 导入jar包到本地库

      有时候有些第三方就是从仓库中下载不下来，配的私服也有问题，就只有手动导入jar包了。Myeclipse 10 可以使用“Myeclipse -- utilities -- Maven4MyEclipse -- import jar to local repository”工具，导入jar包到本地库。eclipse的m2eclipse插件暂时没找到这样的工具。使用maven命令的可以这样：

mvn install:install-file

-DgroupId=com.danga

-DartifactId=memcached

-Dversion=2.0.1

-Dfile=java\_memcached-release\_2.0.1.jar (或者/d:/java\_memcached-release\_2.0.1.jar)

-Dpackaging=jar -DgeneratePom=true

This will add the memcache jar into your local Maven2 repository under groupId com.danga and artifactId memcached, you can then edit your pom.xml adding this dependency.

However, the maven eclipse can not recognize it since it always search from public repository .

安装到私服

    mvn deploy:deploy-file -DgroupId=org.apache.**[Hadoop](http://lib.csdn.net/base/hadoop" \o "Hadoop知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**  -DartifactId=**[Hbase](http://lib.csdn.net/base/hbase" \o "Hbase知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)** -Dversion=1.0 -Dpackaging=jar -Dfile=[path to file] -Durl=[url] -DrepositoryId=[id]

批量导入jar

    直接拷贝文件至/opt/data/nexus/sonatype-work/nexus/storage/pvinsight/org/apache/hadoop/**[Hive](http://lib.csdn.net/base/hive" \o "Hive知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**/hive-exec/0.5.0

    或者

    通过脚本执行 mvn deploy:deploy-file

9. 使用maven发布是，报某个类找不到，原因是某个运行时的类，import了用于**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**的类(但并未使用)，比如junit相关的类。这样，在ide中和maven compile时都不会报错，在打包时，由于junit这样的jar的scope是test类型，因此就不会包含进来，于是就出错了。解决方法就是删除不不要的引用。

### 建立maven的本地代码库服务器，也就是“私服”

Maven的服务器很简单，一个普通的http服务器就行，比如apache，装好apache以后。

将你本地代码库文件夹拷贝到apache安装目录下htpdocs/maven2，windows用户是C:/Documents and Settings/jeffrey/.m2/repository，**[Linux](http://lib.csdn.net/base/linux" \o "Linux知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**下是 /home/Jeffrey/.m2/repository（注意.m2在linux下是隐藏文件夹）

启动apache服务器，在浏览器中输入http://localhost/maven2/repository，应该可以查看到repository的文件目录

编辑项目pom，指定本地的代码库服务器

在project中加入：

    xml 代码

<repositories>

  <repository>

    <id>mylibsid>

    <url>http://localhost/maven2/repositoryurl>

  repository>

repositories>

可以加入多个repository，mvn会按照顺序检索，如果都没有找到，最后会去找官方的服务器

## maven问题汇总

### 常用命令

maven日用三板斧

mvn package   打包(在每个模块的target下生成对应文件)

mvn install        打包，并安装到本地库，如果package失败，报告应用模块找不到，那就试试install

mvn eclipse:eclipse 生成eclipse项目文件

Maven/AntX命令对照表

                   AntX         Maven

构建项目         antx reactor  mvn install

发布项目         antx reactor -P release mvn install -Denv=release

生成eclipse     antx eclipse   mvn eclipse:eclipse

清空项目         antx clean      mvn clean

更新二/三方库        svn up (库目录)     mvn a2m:purge

指定配置文件         antx -u antx.properties mvn -DuserProp=antx.properties

Maven常用命令

命令                 说明

mvn archetype:generate         创建 Maven 项目

mvn compile    编译源代码

mvn test-compile     编译测试代码

mvn test  运行应用程序中的单元测试

mvn eclipse:eclipse 生成 Eclipse 项目文件

mvn eclipse:clean    清除 Eclipse项目文件

mvn package   依据项目生成 jar 文件

mvn install        在本地 Repository 中安装 jar

mvn clean         清除目标目录中的生成结果

mvn site   生成项目相关信息的网站

Maven常用参数

参数         说明

mvn -U     强制更新snapshot类型的插件或依赖库（否则maven只会一天更新一次snapshot依赖)

mvn -o      运行在offline模式，不联网的时候适用

mvn -N     仅在当前项目模块执行命令，关闭reactor

mvn -pl [opentech:module\_name]  在指定的模块上执行命令的，而非全部模块

mvn -ff      在递归执行命令过程中，一旦发生错误就直接退出(fail fast)

mvn -Dxxx=yyy          指定**[Java](http://lib.csdn.net/base/javase" \o "Java SE知识库" \t "http://blog.csdn.net/natian306/article/details/_blank)**全局属性

mvn -Pxxx         应用profile xxx

Maven插件常用参数

命令         说明

mvn -Dmaven.test.skip=true  如果命令包含了test phase，则忽略单元测试（用一次，rp值少10点，自己看着办）

mvn -DdownloadSources=true -Declipse.addVersionToProjectName=true eclipse:eclipse   生成eclipse项目文件，尝试从repository下载源代码，并且生成的项目名包含模块版本（注：如果用公共POM,上述的开关缺省已打开）

mvn dependency:tree -Dverbose    为当前所在模块及所有子模块生成详细的依赖图（包括被仲裁掉的）

### Maven简单故障排除

命令                      说明

mvn -Dsurefire.useFile=false      如果执行单元测试出错，用该命令可以在console输出失败的单元测试及相关信息

set MAVEN\_OPTS=-Xms256m -Xmx512m

-XX:MaxPermSize=128m -XX:ReservedCodeCacheSize=64m Maven内存不够

mvn -X                       maven log level设定为debug在运行

mvndebug        运行jpda允许remote debug

mvn --help        如果不清楚这个命令用法，请和我们联系，马上安排专人伺候～

### 如何查找maven插件的文档

maven所有插件都是maven编译的，在发布的时候，都会为每个插件的配置参数生成一个插件文档：比如我们自己的maven-car-plugin插件，文档在这里：http://repo.alibaba-inc.com/mvn/internal/snapshots/sites/maven-car-plugin/project-reports.html

定位这个文档是有模式可循的：maven generated site -> Project Reports -> Plugin Documentation

定位maven core/plugins的svn:

同上,maven generated site -> Project Information -> Source Repository 按照这个规则去找maven-car-plugin的svn信息吧。

如果要google某个插件的用法，比如dependency插件，请用关键词maven-dependency-plugin.

记得关键词模式是：maven-${pluginname}-plugin

如何查找某个二方库或者三方库jar的pom设置groupId和artifactId？

全文检索的方式查询maven库中是否存在某个jar包，二方库或者三方库：

http://repo.alibaba-inc.com/archiva/index.action

查到后点击可得到pom设置，如：

<dependency>

<groupId>com.alibaba.external</groupId>

<artifactId>ajax.dwr</artifactId>

<version>3.0.0.rc1</version>

</dependency>

淘宝maven库的查询： http://mvnrepo.taobao.ali.com:8081/nexus/index.html#welcome

我们现在不能用淘宝的maven库？setting.xml里有，但是未启用。

### 其他一些问题的解决方案

#### m2eclipse 更新index时间太长的解决办法？

A: 下载index.zip等文件到本地，放到tomcat下并运行tomcat，eclipse打开，设定索引url为http://repo.alibaba-inc.com/mvn/ 然后update index即可。

#### mvn test失败后只显示了失败个数，没看到异常信息

A: 在target/surefire-reports目录下有各测试方法失败的详细信息，查看命令：cat target/surefire-reports/\*.txt

#### 如何引用pom文件上的内容？

通过${project.xxx}引用当前pom文件的任意节点的内容，比如：

${project.packaging} 项目的打包类型，对应<project><packaging>jar</packaging></project>的内容。

另外，一些常用的目录变量：

${basedir} 项目根目录

${project.build.directory} 构建目录，缺省为target

${project.build.outputDirectory} 构建过程输出目录，缺省为target/classes

如何在Maven中为同一个项目指定不同的antx properties文件？

maven-autoconf-plugin可以支持从命令行指定antx.properties。

mvn -DuserProp=/x/y/z/my.properties

That's it !

 更多配置参数参见插件文档：

 http://repo.alibaba-inc.com/mvn/internal/snapshots/sites/maven-autoconf-plugin/config-mojo.html

#### 缺少依赖

有时 build 会因为找不到某些依赖而失败，如下面这段输出：

Missing:

----------

1) com.alibaba.external:jakarta.log4j:jar:1.2.16

  Try downloading the file manually from the project website.

  Then, install it using the command:

      mvn install:install-file -DgroupId=com.alibaba.external -DartifactId=jakarta.log4j -Dversion=1.2.16 -Dpackaging=jar -Dfile=/path/to/file

  Alternatively, if you host your own repository you can deploy the file there:

      mvn deploy:deploy-file -DgroupId=com.alibaba.external -DartifactId=jakarta.log4j -Dversion=1.2.16 -Dpackaging=jar -Dfile=/path/to/file -Durl=[url] -DrepositoryId=[id]

  Path to dependency:

        1) com.alibaba.dragoon:dragoon-common:jar:2.5.0-SNAPSHOT

        2) com.alibaba.external:jakarta.log4j:jar:1.2.16

如果缺少的是二方库( groupId 为 com.alibaba.shared )或三方库( groupId 为 com.alibaba.external )，且确定缺少的依赖在二方库或三方库中是存在的，那出现这个问题很有可能是你的 pom 或 settings 没有配置正确。解决这个问题可以从两个方面入手，首先查看你使用的 settings 是否是 b2b 最新的 settings，最新的 settings 地址为http://svn.alibaba-inc.com/repos/ali\_platform/maven/settings/settings.xml，其次检查你 pom 里面有没有配置 parent，或者运行以下命令，并把输出重定向到文件中：

mvn help:effective-pom -Doutput=output.txt

然后在文件中查找二方库的地址：http://repo.alibaba-inc.com:9091/shared 和三方库的地址：http://repo.alibaba-inc.com:9091/external。

注： 在顶级 pom 下运行的时候最好加上 -N 选项，指示 maven 不递归到当前项目的子模块中。

有的时候配置了正确的 settings，但没有设置 parent，就可能会引起一些三方库找不到，这是因为二方库的地址在settings 里面已经有了配置，而三方库的地址仅在 parent pom com.alibaba:b2b 中配置了，所以如果三方库找不到，那就要在 pom 里设置一个相应的 parent