Maven Build

使用maven-jar-plugin打包

maven-compiler-plugin

- 设置源码和目标的java|jdk|jre版本
- 设置编码格式

```xml
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>

# 或

<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.3</version>
    <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>utf-8</encoding>
    </configuration>
</plugin>


<build>
    <finalName>${project.name}</finalName>
    <plugins>
        <!--<plugin></plugin>-->
    </plugins>
</build>
```

maven-dependency-plugin

- 打包处理单独依赖

```xml
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-dependency-plugin</artifactId>
    <version>3.0.2</version>
    <executions>
        <execution>
            <id>copy-dependencies</id>
            <phase>package</phase>
            <goals>
                <goal>copy-dependencies</goal>
            </goals>
```

```
            <configuration>
                <outputDirectory>${project.build.directory}/lib</outputDirectory>
                <overWriteReleases>false</overWriteReleases>
                <overWriteSnapshots>false</overWriteSnapshots>
                <overWriteIfNewer>true</overWriteIfNewer>
            </configuration>
        </execution>
    </executions>
</plugin>
```

maven-jar-plugin

- 设置打包运行主类和依赖
- 排除重复的文件和文件夹

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-jar-plugin</artifactId>
    <version>3.0.2</version>
    <configuration>
        <archive>
            <manifest>
                <mainClass>${main.class}</mainClass>
                <addClasspath>true</addClasspath>
                <classpathPrefix>lib/</classpathPrefix>
            </manifest>
        </archive>
        <excludes>
            <exclude>etc/</exclude>
            <exclude>jni/</exclude>
            <exclude>log/</exclude>
        </excludes>
    </configuration>
</plugin>
```

maven-resources-plugin

- 打包是将配置文件放在包外

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-resources-plugin</artifactId>
    <version>3.0.2</version>
    <executions>
        <execution>
            <id>copy-resources</id>
            <phase>validate</phase>
            <goals>
                <goal>copy-resources</goal>
            </goals>
            <configuration>
                <outputDirectory>${project.build.directory}/</outputDirectory>
                <resources>
                    <resource>
                        <directory>src/main/resources/</directory>
                        <filtering>true</filtering>
                        <includes>
```

```
                <include>etc/*</include>
                <include>jni/*</include>
            </includes>
            <excludes>
                <exclude>log/*</exclude>
            </excludes>
        </resource>
    </resources>
</configuration>
    </execution>
</executions>
</plugin>
```

使用maven-assembly-plugin打包

使用maven-shade-plugin打包

maven-compiler-plugin

- 设置源码和目标的java|jdk|jre版本
- 设置编码格式

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>

# 或

<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.3</version>
    <configuration>
        <source>1.8</source>
        <target>1.8</target>
        <encoding>utf-8</encoding>
    </configuration>
</plugin>
```

maven-resources-plugin

- 打包是将配置文件放在包外

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-resources-plugin</artifactId>
    <version>3.0.2</version>
    <executions>
        <execution>
            <id>copy-resources</id>
            <phase>validate</phase>
            <goals>
                <goal>copy-resources</goal>
            </goals>
            <configuration>
                <outputDirectory>${project.build.directory}/</outputDirectory>
                <resources>
                    <resource>
                        <directory>src/main/resources/</directory>
                        <filtering>true</filtering>
                        <includes>
                            <include>etc/*</include>
                            <include>jni/*</include>
                        </includes>
                        <excludes>
                            <exclude>log/*</exclude>
                        </excludes>
                    </resource>
                </resources>
            </configuration>
        </execution>
    </executions>
</plugin>
```

maven-shade-plugin

- 设置主类
- 编译版本
- 解决冲突
- 取消证书

```
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-shade-plugin</artifactId>
    <version>2.4.3</version>
    <executions>
        <execution>
            <goals>
                <goal>shade</goal>
            </goals>
            <configuration>

                <transformers>
                    <transformer
implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                        <manifestEntries>
                            <Main-Class>${main.class}</Main-Class>
```

```xml
                        <X-Compile-Source-JDK>${java.version}</X-Compile-Source-
JDK>
                        <X-Compile-Target-JDK>${java.version}</X-Compile-Target-
JDK>
                    </manifestEntries>
                </transformer>
                <transformer
implementation="org.apache.maven.plugins.shade.resource.AppendingTransformer">
                    <resource>META-INF/spring.handlers</resource>
                </transformer>
                <transformer
implementation="org.apache.maven.plugins.shade.resource.AppendingTransformer">
                    <resource>META-INF/spring.schemas</resource>
                </transformer>
            </transformers>

            <filters>
                <filter>
                    <artifact>*:*</artifact>
                    <excludes>
                        <exclude>etc/**</exclude>
                        <exclude>jni/**</exclude>
                        <exclude>log/**</exclude>
                        <exclude>META-INF/*.SF</exclude>
                        <exclude>META-INF/*.DSA</exclude>
                        <exclude>META-INF/*.RSA</exclude>
                    </excludes>
                </filter>
            </filters>
        </configuration>
    </execution>
</executions>
</plugin>
```