MavenAndClasspath

This page describes why Maven coordinates and Java class package names need to be co-ordinated.

Maven classpath

Maven identifies jars using the coordinate triple {groupId, artifactId, version}. It assumes that jars with the same pair {groupId, artifactId} are different versions of the same jar, and will ensure that only one instance of the jar is added to the classpath (in general the one with the highest version).

Jars with different {groupId, artifactId} pairs are treated as entirely different jars, even if they contain identical classes. Maven does not examine the contents of jars; it relies on the {groupId, artifactId} pair to determine whether different jars have the same or different classes in them.

Java classloader

The Java runtime system only allows a single instance of a class to exist in a given classloader. Classes are uniquely identified using the package name and class name.

For example, the class `Utils` in the package `org.apache.commons.example` can only appear once in a classloader.

The classloader uses the classpath to find the classes needed by an application.

If the classpath contains more than one jar which contains the above Utils class, then the JVM will load only one of the instances.

If the class instances in the different jars are identical, then this is not a problem, but if there are two different versions of the class, the classloader may choose the wrong one. And different JVMs may process the classpath in a different order.

To ensure that the expected class is loaded, it is vital that the classpath only contain a single instance of each different class. This is normally achieved by grouping classes in jars according to their package names, and ensuring that there is only one instance of each such jar on the classpath.

==Maven coords and package names==

When using Maven to construct the classpath, it is therefore essential that Maven knows which jars hold different versions of the same classes and which jars have completely different classes.

Therefore, it is vital that two jars with the same {groupId, artifactId} pair use the same package name. Likewise, it is vital that two jars with the same package name use the same {groupId, artifactId} pair.

There must be a 1-1 correspondance between the Maven {groupId, artifactId} pair and the Java package name.

If either/both of the Maven groupId or artifactId is changed, the package name must be changed. Otherwise, Maven may add more than one copy of a class to the classpath (in different jars) causing unpredictable behaviour.

Likewise, a change of package name (e.g. for a binary break) must be accompanied by a change of either Maven groupId or artifactId. Otherwise, Maven won't be able to to add the original package name to the classpath, breaking code that relies on the original package name.