



addjars-maven-plugin

Add dependency to local jar files in maven projects

 Search projects[Project Home](#)[Wiki](#)[Issues](#)[Source](#)[Export to GitHub](#) Search

★ UsagePage

AddJars Maven Plugin usage

Updated Dec 10, 2012 by y.kary...@gmail.com

Prerequisites

The plugin requires Maven version 3.0.3 or higher. Unfortunately, it doesn't work with earlier maven versions.

Simple usage

Suppose your project's classpath needs to include all jar files located in the `lib` directory of your project (including subdirectories). To achieve this you should put the following declaration to the list of plugins in your `pom.xml`:

```

<plugin>
  <groupId>com.googlecode.addjars-maven-plugin</groupId>
  <artifactId>addjars-maven-plugin</artifactId>
  <version>1.0.5</version>
  <executions>
    <execution>
      <goals>
        <goal>add-jars</goal>
      </goals>
      <configuration>
        <resources>
          <resource>
            <directory>${basedir}/lib</directory>
          </resource>
        </resources>
      </configuration>
    </execution>
  </executions>
</plugin>

```

Advanced usage

Apart from specifying the directory of the jar resources to be added, you can also specify the dependency scope and lists of includes and excludes:

```

<plugin>
  <groupId>com.googlecode.addjars-maven-plugin</groupId>
  <artifactId>addjars-maven-plugin</artifactId>
  <version>1.0.5</version>
  <executions>
    <execution>
      <goals>
        <goal>add-jars</goal>
      </goals>
      <configuration>
        <resources>
          <resource>
            <directory>${basedir}/lib</directory>
            <includes>
              <include>**/*.jar</include>
            </includes>
            <excludes>
              <exclude>${basedir}/lib/runtime/**/*.jar</exclude>
            </excludes>
          </resource>
          <resource>
            <directory>${basedir}/lib/runtime</directory>
            <scope>runtime</scope>
          </resource>
        </resources>
      </configuration>
    </execution>
  </executions>
</plugin>

```

In the given example, all jar files located in the `lib` directory, except for the ones located in `lib/runtime`, will be added to your project with

scope compile. The other files (located in lib/runtime) will be added with scope runtime.

The includes and excludes have the same meaning as in the main resources section (see <http://maven.apache.org/pom.html#Resources>).

IDE, addjars-maven-plugin, and autocompletion

If you use NetBeans (not sure about other IDEs) and the addjars-maven-plugin configuration is used in a module of packaging jar or war, you will probably have problems with autocompletion. That's because NetBeans doesn't consider the plugin configuration.

Fortunately, there is a solution. The project should have multi-module setup and consist of two modules at least.

First of the modules should be of packaging pom and include the addjars-maven-plugin configuration:

```
<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>

  <parent>
    <groupId>my.group.id</groupId>
    <artifactId>parent</artifactId>
    <version>my-project-version</version>
  </parent>

  <artifactId>lib</artifactId>
  <packaging>pom</packaging>

  <build>
    <plugins>
      <plugin>
        <groupId>com.googlecode.addjars-maven-plugin</groupId>
        <artifactId>addjars-maven-plugin</artifactId>
        <version>1.0.5</version>
        <executions>
          <execution>
            <goals>
              <goal>add-jars</goal>
            </goals>
            <configuration>
              <resources>
                <resource>
                  <directory>${basedir}/lib</directory>
                </resource>
              </resources>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>
```

Other (probably jar or war) module(s), which depend on the third party jars, should refer to the first module in their dependencies sections:

```
<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>

  <parent>
    <groupId>my.group.id</groupId>
    <artifactId>parent</artifactId>
    <version>my-project-version</version>
  </parent>

  <artifactId>webapp</artifactId>
  <packaging>war</packaging>

  ...

  <dependencies>
    <dependency>
      <groupId>${project.groupId}</groupId>
      <version>[${project.version}]</version>
      <artifactId>lib</artifactId>
      <type>pom</type>
    </dependency>
    ...
  </dependencies>
```

```
</project>
```

Comment by [rogerio....@gmail.com](#), Jul 12, 2012

The 1 and 2 tags, shouldn't be 1a and 2b for the auto-completion step? I used this configuration and works great! Thank you for this plugin, a lot! 1: <artifactId>lib</artifactId> 2: <type>pom</type>

1a: <artifactId>\${project.artifactId}</artifactId> 2a: <type>\${project.packaging}</type>

Comment by [K...@navidao.com](#), Sep 5, 2012

It works great when I doing mvn compile, but when I try to package it with maven shade-plugin.

I got the error like this:

org.apache.maven.plugins:maven-shade-plugin:1.7.1:shade (default) on project testing: Error creating shaded jar: invalid LOC header (bad signature)

Is that because the jars that installed on local rep didn't have proper signature?

How can I get resolved?

Comment by [carlos.alonso.gonzalez](#), Oct 1, 2012

The plugin is fantastic but there is one problem, the plugin doesn't work with tomcat-maven-plugin goal tomcat:run.

Web application with maven comand: "mvn clean tomcat:run", the error result is:

```
Failed to execute goal com.googlecode.addjars-maven-plugin:addjars-maven-plugin:1.0.4:add-jars (default) on project g
at org.apache.maven.lifecycle.internal.MojoExecutor?.execute(MojoExecutor?.java:217) at org.apache.maven.lifecycle
Caused by: org.apache.maven.plugin.MojoExecutionException?
at com.googlecode.addjars.mojo.AddJarsMojo?.execute(AddJarsMojo?.java:83) at org.apache.maven.plugin.DefaultBuild
Caused by: java.lang.UnsupportedOperationException?
at java.util.Collections$UnmodifiableCollection?.add(Collections.java:1018) at com.googlecode.addjars.mojo.AddJar
```

Comment by project member [v.kary...@gmail.com](#), Dec 9, 2012

Thanks everyone for your comments. Was not aware of them until today.

As for the issues you mentioned, I'm aware of them and don't see an easy way to fix them. The plugin is a hack in nature, and the hack often conflicts with hacks used in other plugins :)

As workaround, I suggest moving the plugin declaration to a separate pom (of packaging 'pom'). Just like the last section of this page suggests (see "IDE, addjars-maven-plugin, and autocompletion").

Comment by [tarun.gu...@technogica.com](#), Feb 15, 2013

To make it work with Eclipse 3.8 please see my reply to a post here : <https://groups.google.com/d/msg/addjars-maven-plugin/G4SC1mDXSfE/OqUMdsDJm6wJ>

Comment by [ra...@omedix.com](#), Nov 21, 2013

In My jar files i getting extensions .lastUpdated... Any help

Comment by [stepha...@imagemattersllc.com](#), Nov 27, 2013

I couldn't get the plugin working with the configuration above. I got a meven error indicating that resources are missing or invalid.

```
<plugins>
```

```
<plugin>
```

```

<groupId>com.googlecode.addjars-maven-plugin</groupId> <artifactId>addjars-maven-plugin</artifactId> <version>1.0.5</version>
<configuration>

    <resources>

    <resource>

        <directory>${basedir}/lib</directory>

    </resource>

    </resources>

</configuration>

<executions>

<execution>

    <goals>

        <goal>add-jars</goal>

    </goals>

</execution>

</executions>

</plugin>

</plugins>

```

Configuration is put outside the executions tag.

Comment by stepha...@imagemattersllc.com, Nov 27, 2013

Again the script to import dependency does not work:

```

<dependency>
    <groupId>${project.groupId}</groupId>
    <version>[${project.version}]</version>
    <artifactId>lib</artifactId>

```

It should be instead:

```

<dependency>
    <groupId>${project.groupId}</groupId>
    <version>${project.version}</version>
    <artifactId>${project.artifactId}</artifactId>
    <type>${project.packaging}</type>
</dependency>

```

```

<type>pom</type>

```

```

</dependency>

```

Enter a comment:

Hint: You can use [Wiki Syntax](#).

Submit

[Terms](#) - [Privacy](#) - [Project Hosting Help](#)

Powered by [Google Project Hosting](#)