



## MAVEN3 TOOL

by: Dandár Gábor and Lajtos Máté

# Plugin and archetype for Maven3

Brillien provides its own plugin and archetype to help development. Using these, one can create and manage a project that is compatible with Brillien. This includes creating the project structure, interpreting and building the source code, and publishing it into an available Brillien distribution.

## Installing the archetype and the plugin

Download the necessary files, which are available in Brillien's SVN repository (*source/maven* folder), or by clicking on the Maven3 tools link on the Brillien website. After the download, the *brillien-archetype* and the *brillien-maven-plugin* folders contain the Maven3 tools which have to be installed. The following command in both of these folders have to be executed:

```
mvn install
```

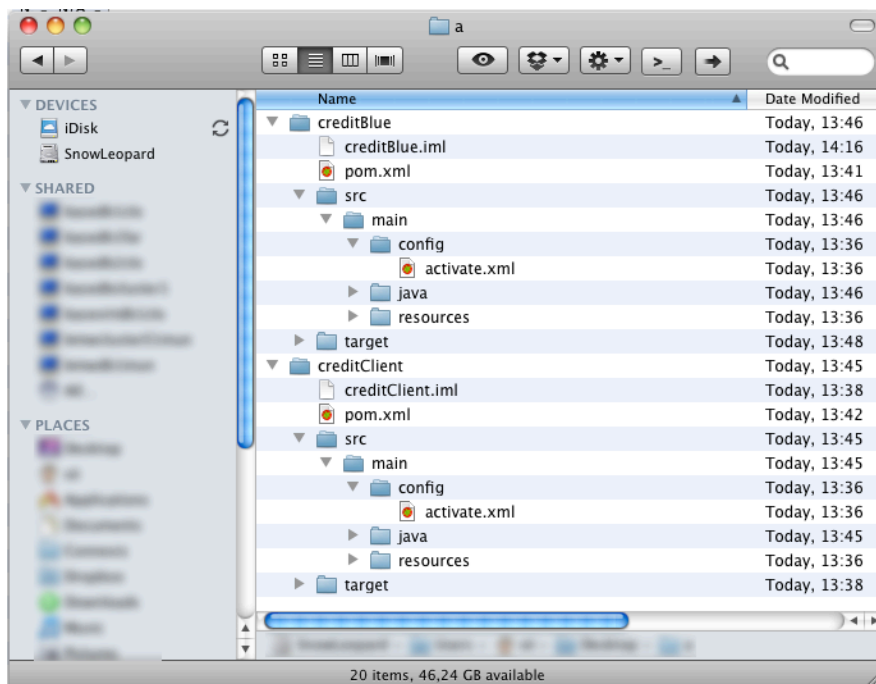
After the successful installation, you can start using these tools.

## Starting a new project

In case of a Brillien-based development, the Maven3 *archetype* creates the appropriate project structure and creates the necessary *pom.xml*. You can define a new Brillien project with the following operation:

```
mvn archetype:create
-DgroupId=[your project's group id]
-DartifactId=[your project's artifact id]
-DarchetypeGroupId=com.vii.brillien
-DarchetypeArtifactId=brillien-archetype
-DarchetypeVersion=11.04
```

You have to specify the *groupId* and the *artifactId*. After the execution, you'll get a folder structure just like this (source: user's manual of Brillien, Tutorial 0):



The development of the project can be started with any IDE which supports Maven3. The *pom.xml* file of the project contains the following:

```
<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>
  <groupId>com.acme.product</groupId>
  <artifactId>product</artifactId>
  <version>1.0-SNAPSHOT</version>
  <dependencies>
    <dependency>
      <groupId>com.vii.brillien</groupId>
      <artifactId>core</artifactId>
      <scope>provided</scope>
      <version>11.04</version>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-dependency-plugin</artifactId>
        <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>
              <excludeScope>provided</excludeScope>
            </configuration>
          </execution>
        </executions>
      </plugin>
      <plugin>
        <groupId>com.vii.brillien</groupId>
        <artifactId>brillien-maven-plugin</artifactId>
        <version>11.04</version>
        <executions>
          <execution>
            <id>brillienbuild</id>
            <phase>package</phase>
            <goals>
              <goal>brillienbuild</goal>
            </goals>
          <goal>deploy</goal>
          </goals>
          <configuration>
            <brillienHome>path to Brillien home directory</brillienHome>
          </configuration>
        </executions>
      </plugin>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>2.0.2</version>
      </plugin>
    </plugins>
  </build>
</project>
```

```
        <configuration>
          <source>1.6</source>
          <target>1.6</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

The plugin defines two lifetime tasks:

```
brillien:brillienbuild : csomagolás
brillien:deploy       : publikáció
```

After compiling the source code and executing the *brillien:brillienbuild* command, the archive file Brillien module will be created and copied into the *src/main/target/brilliendist* folder, from where the *brillien:deploy* command will copy it into the activator folder of the local Brillien distribution. For this, the *<brillienHome>* tag in the *pom.xml* file must be given a value.

The *mvn package* command will execute every necessary operation on the source code of our project, from compiling to publishing.

## Building the Brillien

Download the source of Brillien from the SVN repository (<https://brillien.googlecode.com/svn/trunk/>), go to the *source/src* folder, then execute the following command:

```
mvn install
```

During the execution, the Maven compiles the source codes and publishes the created *jar* files in the local Maven2 repository. If a whole binary distribution is needed, execute the following command:

```
mvn package assembly:assembly
```

After executing this, the *Brillien-[version\_number]-distribution.zip* file will be created in the *src/target* folder. The binary distribution on the Brillien website is created like this, too. After extracting this, executing the appropriate command file, the server can be started.