

Artifact tool

Introduction

Daisy makes intensive use of the [artifact](#)¹ concept. Both [Maven](#)² and the [Daisy Runtime](#)³ make use of these artifacts.

Since there are a lot of different Maven POMs and Daisy Runtime classloader definitions, it is useful to have some help in managing these. For example when you want to find out which projects use commons-collections or update all of them to a newer version. Or you want to augment the version number of all Daisy artifacts.

Rather than doing this manually, which is boring and error-prone, we have a tool called the "artifacter" which does this.

General usage instructions

First compile the artifacter tool:

```
NOTE
<daisy-src> refers to the root of the Daisy source tree
```

```
cd <daisy-src>
cd tools/artifacter
maven jar:jar
```

Then the typical usage is:

```
cd <daisy-src>

./tools/artifacter/target/artifacter-dev <options>
```

You should usually invoke the tool from the root of the Daisy source tree, since it will work on all files found recursively from the current working directory.

Finding out which projects use a certain artifact

This lists all files where a certain artifact is used.

Note that the argument of the `-f` parameter is `<groupId>:<artifactId>`

```
./tools/artifacter/target/artifacter-dev -f daisy:daisy-repository-api
```

Upgrading a dependency

For example, let's move to a new version of Jetty:

-
1. [daisy:490-cd](#) (Artifact)
 2. <http://maven.apache.org>
 3. [daisy:467-cd](#) (Daisy Runtime)

```
./tools/artifacter/target/artifacter-dev -u jetty:org.mortbay.jetty:16.2.3
```

Upgrading a project version number

Within Daisy, this is only useful for upgrading the Daisy version number.

It upgrades version numbers not only of dependencies, but also of the project definitions themselves.

It works on all artifacts within a certain group.

Example usage:

```
./tools/artifacter/target/artifacter-dev -g daisy:2.8
```

Renaming an artifact

Less common, but it can happen (this was mainly used in the early Daisy days).

```
./tools/artifacter/target/artifacter-dev  
-r -o daisy:daisy-repository-api -n daisy:daisy-repository-api2
```

Creating a repository of all dependencies

This creates a Maven-style repository containing all dependencies from the runtime classloader definitions. This is used by the binary distribution build script.

```
./tools/artifacter/target/artifacter-dev -c -s <source-repo> -d <dest-repo>
```

The <source-repo> is typically ~/.maven/repository

The <dest-repo> is typically a new empty directory.

Copy project dependencies

This copies the dependencies of a certain Maven project to a Maven-style local repository. This is used by the binary distribution build script.

```
./tools/artifacter/target/artifacter-dev -l -s <source-repo> -d <dest-repo>
```

Copy artifact

This copies the artifact build by a certain Maven project to a Maven-style local repository. This is used by the binary distribution build script.

```
./tools/artifacter/target/artifacter-dev -a -s <source-repo> -d <dest-repo>
```