

MavenInfo

Overview

The apache JDO project uses [maven](#) for build. The current maven version is 1.0.2; you can download it from [here](#).

Maven Groups and Artifacts

Apache JDO contains two distinct types of projects: api11 and api20 build the javax.jdo contents that are officially documented in the JSR-12 and JSR-243 specifications. The group id for these two projects is javax.jdo. The artifact ids are jdo-api11 and jdo-api20.

The second type of project includes the Technology Compatibility Kits, Reference Implementation, and sample integrations of JDO. These are not documented in the specifications but rather are developed as ordinary open source projects. The group id for these projects is org.apache.jdo. The artifact ids are jdo1-ri, jdo1-tck, jdo-btree, jdo2-tck, jdo2-fostore, jdo2-core, jdo2-enhancer, jdo2-runtime, and jdo2-query.

Getting Started

The maven page provides a [GettingStarted](#) description. Another useful link is the [user guide](#).

You need at least two environment variables:

```
MAVEN_HOME root directory of your maven installation
JAVA_HOME root directory of your JDK installation
```

You might need to extend the JVM heap size when running some maven goals. An easy way is defining the environment variable as

```
MAVEN_OPTS=-Xmx512m
```

Once you installed maven you could try to build the apache jdo maven project. You find the latest zip at [SubversionRepository](#). Unzip the file, cd to the directory jdo and then call

```
maven -Dgoal=build multiproject:goal
```

This builds the sub-projects api11, btree, ri11, and tck11. It will take a while, since this compiles and tests the 4 sub-projects. The following command builds everything including the documentation:

```
maven multiproject:site
```

Access through a proxy

Maven allows to set a HTTP proxy host and port so that it can tunnel through a HTTP Proxy if you are behind a firewall. You find details [here](#).

You can create a file \${user.home}/build.properties with the following two lines:

```
maven.proxy.host = your.proxy.name
maven.proxy.port = 3128
```

Please use the regular http proxy settings in case you want to access the repository from the Sun network (SWAN).

The btree subproject accesses the.netbeans cvs repository to download the btree sources. This might not work if you are behind a firewall that blocks the cvs port [see](#). If you are inside the Sun network (SWAN) you can use the alternative CVSROOT.

```
CVSROOT=:pserver:anoncvs@cvsnetsun.com:/cvs
```

Please have a look at the file project.properties of the btree sub-project.

IDE plugins

There is a nice maven plugin called Mevenide for Eclipse and Netbeans. You find details [here](#).