

Releasing Studio

Since we are using Nexus for releases the release process is as follows (see also <http://www.apache.org/dev/publishing-maven-artifacts.html#staging-maven>).

0. Verify Your Maven Settings

Several things need to be correctly configured in your '*settings.xml*' file located at '*~/m2/settings.xml*'.

Here's a sample file:

```
<settings>
  <servers>
    <!-- To publish a snapshot -->
    <server>
      <id>apache.snapshots.https</id>
      <username>[Your Apache ID]</username>
      <password>[Your Apache Password]</password>
    </server>
    <!-- To stage a release -->
    <server>
      <id>apache.releases.https</id>
      <username>[Your Apache ID]</username>
      <password>[Your Apache Password]</password>
    </server>
  </servers>

  <profiles>
    <profile>
      <id>apache-release</id>
      <properties>
        <!-- To sign a release -->
        <gpg.keyname>[Your Code Signing Key ID]</gpg.keyname>
        <gpg.passphrase>[Your Code Signing Key Password]</gpg.passphrase>
      </properties>
    </profile>
  </profiles>
</settings>
```

1. Test the Project

```
$ mvn release:prepare -DdryRun=true
```

Be aware that this phase will ask you about the release version of Studio modules.
We use a release number scheme that combines a version number and the date of the release:

```
[Version using 'X.Y.Z' format].v[Date using 'YYYYMMDD' format]
```

An example of this release number scheme is '*2.0.0.v20120111*' (notice the '*.v*' string between the version and the date).

2. Deploy a Snapshot

```
$ mvn deploy
```

This is useful to verify your settings in *~/m2/settings.xml* (Nexus password and GPG key)

3. Prepare the Release

First, clean the previous test release attempt with:

```
$ mvn release:clean
```

Next, edit the root *'pom.xml'* file to comment the *'repository'* and *'application'* profiles, as we don't want these modules (and their sub-modules) to be released in the Maven repository.

Also, edit the *'pom.xml'* file of the *'Apache Directory Studio Libraries Plugins'* module and remove the code related to skipping the deployment of 3rd party plugins:

```
<build>
  <plugins>
    <!-- Skip deployment of 3rd party plugins. This causes problems when
         deploying snapshots because the 3rd party plugins don't have a
         snapshot version. -->
    <plugin>
      <artifactId>maven-deploy-plugin</artifactId>
      <configuration>
        <skip>true</skip>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Then, prepare the release with:

```
$ mvn release:prepare
```

This creates a tag here: <http://svn.apache.org/viewvc/directory/studio/tags/>

4. Stage the Release

```
$ mvn release:perform
```

This deploys the release to a staging repository. Go to <https://repository.apache.org/index.html#stagingRepositories> and close the staging repository.

5. Deploy the Site

```
$ cd target/checkout
$ mvn site-deploy
```

This creates and deploys the site via ssh to people.apache.org/www/directory.apache.org/studio/gen-docs.

Edit the people.apache.org/www/directory.apache.org/studio/gen-docs/index.html file and add the deployed site to the list.

6. Generate the distribution packages

Move into the *'application'* sub-module:

```
$ cd application
```

Edit the *'pom.xml'* files of this sub-modules and all other (sub-)sub-modules to fix the parent version which still refers to the old version number (it was not modified during the release process because the *'application'* profile was commented):

```
$ text-editor-command pom.xml */pom.xml
```

Run the following command to generate all the distribution packages:

```
$ mvn clean install -Prelease
```

Now, you have all the distribution packages available at *'../target/release'*:

```
$ ls -l ../target/release
```

7. Publish Distribution Packages

```
$ cd ../target/release  
$ scp * people.apache.org:public_html/studio-<version>/
```

8. Vote

Start a 72h vote at the dev mailing list.

9. Release

If the vote succeeds Studio project can be released.

Go to <https://repository.apache.org/index.html#stagingRepositories> and release the staging repository so all artifacts are published to Maven central.

Move the distribution packages to the dist SVN repository: [https://dist.apache.org/repos/dist/release/directory/studio/dist/\\${version}](https://dist.apache.org/repos/dist/release/directory/studio/dist/${version})

Move the sources releases to the dist SVN repository: [https://dist.apache.org/repos/dist/release/directory/studio/\\${version}](https://dist.apache.org/repos/dist/release/directory/studio/${version})