

FulcrumMaven2Migration

Maven2 Release Procedure

The Fulcrum Parent POM defines different profiles

- rc (Release Candidate)
- release (Release)

The 'rc' profile copies the artifacts to

people.apache.org/www/turbine.apache.org/builds/turbine/fulcrum/\${fulcrum.componentid}/\${fulcrum.release.version}/\${fulcrum.rc.version}/staged

whereas the 'release' profile copies the artifacts to

people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository

A. Infrastructure Setup (one time)

A.1 Get Familiar with Maven

Reading the Maven documentation is a good start - [Releasing A Maven Project](#).

A.2 Signature Keys

You need to sign the released artifact using your PGP key which implies

- you generated one
- it is uploaded to a key server
- it is signed on a key signing party
- it is found in the KEYS file

A.3 Maven Installation

Currently Maven 2.0.10 or better is required to run the release process

A.4 Maven Server Settings

It is a good idea to check your *settings.xml* that it contains the corresponding server entries for the repositories and websites defined in the fulcrum parent pom. If you have a different user name on your box than on *people.apache.org* your local user name will be used for authentication. Therefore the authentication will fail and after a few retries you are locked out from *people.apache.org* for a day - which in turn makes cutting a release rather lengthy. In theory you could also provide your password in the server section below but this is unsafe and did not work in my case.

```
<servers>
  <server>
    <id>apache.releases</id>
    <username>YOUR_APACHE_USERNAME</username>
    <filePermissions>664</filePermissions>
    <directoryPermissions>775</directoryPermissions>
  </server>
  <server>
    <id>apache.website</id>
    <username>YOUR_APACHE_USERNAME</username>
    <filePermissions>664</filePermissions>
    <directoryPermissions>775</directoryPermissions>
  </server>
  <server>
    <id>apache.snapshots</id>
    <username>YOUR_APACHE_USERNAME</username>
    <filePermissions>664</filePermissions>
    <directoryPermissions>775</directoryPermissions>
  </server>
</servers>
```

B. Release Preparation

B.1 Check Your Project

Here you find a list of reminders what to double-check before cutting a release candidate

- Make sure that you are not referencing any SNAPSHOT version (dependencies and plugins) otherwise the release will fail
- Make sure that you update the *xdocs/changes.xml* file to reflect your release
- Make sure that all your files contain a ASF licence header (look at the *RAT* report when in doubt, generated with *mvn site*)
- Make sure that the LICENSE and NOTICE files are present and correct - especially that the year is correct in the NOTICE file.
- Make sure that you make a test build using a clean checkout
- Make sure that there are no major bugs in JIRA
- Make sure to add all missing committers and contributors

B.2 Prepare Your Version Number

A guideline regarding version numbering can be found <http://commons.apache.org/releases/versioning.html>.

B.3 Prepare Your Maven Variables

During the staging process a directory is created on *people.apache.org* based on the content of the following Maven variables

- `fulcrum.release.version` is a duplicate of the pom version
- `fulcrum.rc.version` is the current number of your release candidate

That could look like the following snippet taken from *yaafi-crypto*

```
<properties>
  <fulcrum.release.version>1.0.0</fulcrum.release.version>
  <fulcrum.rc.version>RC2</fulcrum.rc.version>
</properties>
```

B.3 Define Staging Site for Release Candidate

Also ensure that the `rc` profile will stage the site (rather than update the Fulcrum component website). See [COMMONSSITE-26](#) for background.

```
<profiles>
  <profile>
    <id>rc</id>
    <distributionManagement>
      <!-- Cannot define in parent ATM, see COMMONSSITE-26 -->
      <site>
        <id>apache.website</id>
        <name>Fulcrum Release Candidate Staging Site</name>
        <url>${fulcrum.deployment.protocol}://people.apache.org/www/turbine.apache.org/fulcrum/fulcrum-yaafi-
crypto</url>
      </site>
    </distributionManagement>
  </profile>
</profiles>
```

B.5 Prepare Your Assembly Descriptors

If you are declaring/using your own assembly descriptors make sure that they are not using `${version}` but `${fulcrum.release.version}` - there seems to be an odd bug which results in an incorrectly expanded version string, e.g. `fulcrum-yaafi-crypto-2.4.1-src`. If in doubt look at `./src/assembly/src.xml`.

B.6 Dry Run Your Maven Release

Check that your poms will not lose content when they are rewritten during the release process.

- `mvn release:prepare -DdryRun=true`
- Diff the original file `pom.xml` with the one called `pom.xml.tag` to see if the license or any other info has been removed. **This has been known to happen if the starting `<project>` tag is not on a single line.** The only things that should be different between these files are the `<version>` and `<scm>` elements. Any other changes, you must backport yourself to the original `pom.xml` file and commit before proceeding with the release.
- Remember to do `'mvn release:clean'` before you start the real release process.

C. Run The Release Process

In Commons land there are currently the following conventions

- we use the "real" version number for the RC, e.g. "1.0.6" instead of "1.0.6-RC1"
- we use the "real" SVN tag, e.g. "FULCRUM_YAAFI_CRYPTO_1_0_6" instead of "FULCRUM_YAAFI_CRYPTO_1_0_6_RC1"
- if the RC is accepted the generated artifacts are copied to the M2 repo - so the need to have the "real" version and tag

C.1 Stage Artifacts and Site

1. `mvn -PrC release:prepare`
2. `mvn -PrC release:perform`

When the release plugin prompts for the tag name, choose the final name (for example, `FULCRUM_YAAFI_CRYPT0_1_0_6`).

You may experience memory problems with newer Maven versions. If that happens, set the environment variable

```
export MAVEN_OPTS="-Xmx512m"
```

C.2 Remove SHA1 and MD5 Fingerprints of PGP Signatures

During the release process some unwanted fingerprints of all ASCII-armored files are generated, e.g. "fulcrum-yaafi-crypto.jar.asc.md5" and "fulcrum-yaafi-crypto.jar.asc.sha1". It is a good idea to remove them manually for the time being.

C.3 Fix permissions

For some reason, the site deployed to the staging location will only have group write permissions on the lowest directory level. So it might be necessary to fix the group permissions on you newly created site tree so that others will be able to add new versions later. So like e.g.

```
chmod -R g+w /www/turbine.apache.org/builds/fulcrum/fulcrum-yaafi-crypto
```

D. Vote

D.1 Send Out The Vote

Below you find a vote template to save you some time ...

```
Tag:

https://svn.apache.org/repos/asf/turbine/fulcrum/tags/${TAG}/

Site:

http://turbine.apache.org/builds/fulcrum/YOUR_PROJECT/${fulcrum.release.version}/${fulcrum.rc.version}/site/index.html

Binaries:

http://turbine.apache.org/builds/turbine/fulcrum/${project.artifactId}/${fulcrum.release.version}/${fulcrum.rc.version}/staged/${project.groupId}/${project.artificatId}/${fulcrum.release.version}/

[ ] +1 release it
[ ] +0 go ahead I don't care
[ ] -1 no, do not release it because
```

D.2 React on the Vote

If vote fails, undo and redo previous steps as appropriate (make sure SVN tag is deleted before recreating). And rest assured that most releases need more than one release candidate ... 😊

E. Go live

E.1 Copy to "dist"

Copy distributions to <http://www.apache.org/dist/> area which is located on `people.apache.org` under `/www/www.apache.org/dist/turbine/fulcrum`. The content is also automatically copied to `archive.apache.org`

E.2 Copy Artifacts from Staging

The Maven Stage plugin copies the artifacts (from you release candidate) to the M2 sync directory (`people.apache.org/repo/m2-ibiblio-rsync-repository`).

The following example was used to do the copy the artifacts for `fulcrum-yaafi-crypto-1.0.6` based on RC4

```
mvn stage:copy \
-Dsource="http://turbine.apache.org/builds/turbine/fulcrum/fulcrum-yaafi-crypto/1.0.6/RC4/staged" \
-Dtarget="scp://people.apache.org/www/people.apache.org/repo/m2-ibiblio-rsync-repository" \
-DtargetRepositoryId=apache.releases \
-Dversion=1.0.6
```

E.3 Deploy the Site

Run `mvn site-deploy` to deploy the site - please note that you are deploying the site of the next development snapshot. If you prefer to deploy the actual release site, make sure you keep the directory `target/checkout` and do the deployment from there.

F. Celebrate

F.1 Send an Announcement

Send an announcement to the developer and user mailing list. Since you have dutifully updated your `changes.xml` you can generate the announcement in `./target/announcement/`. Like in the case of the site deployment, this works best if you keep the checked-out project after completing the `release:perform` step.

```
> mvn changes:announcement-generate
```

Outdated Fulcrum M2 Migration Guidelines

We decided to migrate from M1 to M2 not only because M1 is a bit outdated but also because doing a proper release is much easier using M2.

How To Make a M2 release

- Check that the up-to-date `pom.xml` features a SNAPSHOT version, e.g. `1.0.2-SNAPSHOT`
- Run `"mvn clean release:clean"` to remove any temporary files
- Run `"mvn -Prelease release:prepare"` and provide a tag name, e.g. `"FULCRUM_EMAIL_1_0_2"`
- Run `"mvn -Prelease release:perform"`
- Go to `./target/checkout` directory
- Run `"mvn -Prelease clean site site:deploy"` to publish the website
- Run `"mvn changes:announcement-generate"` to generate the announcement in `./target/announcement`