Publishing a Release

The release process follows in general the Apache guidelines for Release-Publishing. When migrating from svn to git the released versions went into git tags - this is fine, but remember, that to be able to have a maintenance releases you could easily create a new release branch. For a new release a tag could be created containing 'candidate', which will be later (after successfull voting) be deleted and renamed (see below).

Find another helpful description of some details of the process here (dbo release instructions).

Git Repositories

- Check with

  Locale Git environment check

  ```
  git config -l
  ```

  that **user.email** is the **apache.org** user e-mail address.
- **Prepare a RC for voting:** Start the Maven Release Process. Assert that you are on the **master/trunk/main branch** (check with git status or git branch)! Following we assume as an example performing a release turbine-parent pom component. Maven release:prepare adds by default ***a tag*** <project-artifact>-<version> = turbine-parent-9. We may want to add to the tag name a postfix "**-candidate**". If the voting process is done, and it is successfull, we have to rename the tag or if we want rather to have a *branch,* we name this new branch exactly like the released component to avoid to have a duplicate tag name (which is actually just a named commit). In this case we would keep the candidate tag name or delete it at all (you may want to check all of this with -DdryRun=true).

  Maven Release

  ```
  mvn release:prepare -Papache-release -Dtag=turbine-parent-9-candidate
  // N.B. mvn release:branch seems not very appropriate,
  mvn release:perform
  ```

  Find more details about maven release for *multi module projects* here (Fulcrum Build)
- **After voting is successfull, do one of the following steps:**
  - **Rename tag for Fulcrum Component foo**

    Rename git tag after voting

    ```
    git checkout <foo>-<version>-candidate
    // .. output ..
    Note: switching to '<foo>-<version>-candidate'.
    You are in 'detached HEAD' state. .,,

    // local: add new tag name
    git tag <foo>-<version><foo>-<version>

    // delete locally candidate tag
    git tag -d <foo>-<version>-candidate

    // remote: two in one
    // add new tag and delete old tag (colon prefix is a shortcut for delete)
    git push origin <foo>-<version> :<foo>-<version>-candidate
    // .. output ..
    // To https://gitbox.apache.org/repos/asf/<foo>.git
    // - [deleted]         <foo>-<version>-candidate
    // * [new tag]         <foo>-<version> -> <foo>-<version>

    // others have to do this also
    git pull --prune --tags origin master
    ```
  - or **Create a release branch** a We assume, that the release is created from the master/main/trunk branch and a release branch will be created later for a maintenance release. **Explicit git commands for this (with checks):**

Release branch

```
git branch -a
git tag -l
// tag name = turbine-parent-<version>-candidate
git checkout turbine-parent-<version>-candidate
// create a branch from tag commit
git checkout -b turbine-parent-<version>
git push -u origin turbine-parent-<version>
```