

# ReleaseProcess

## Apache Tomcat Release Process

### Introduction

This is written primarily for Apache Tomcat release managers although it may also be of interest to anyone looking to validate and/or replicate the release process. This page uses Tomcat 9 as an example but the same process applies to later versions as well. Please ensure that you have an environment that is [set up properly for a release build](#), plus that you are using the latest versions of toolchain components (specifically, the JDK and ant).

### Preliminary checks

- Check that the version numbers have been incremented after the previous release, as expected. (e.g. <https://github.com/apache/tomcat/commit/cec14f9c33af9da271c9681bf6b343c61b6d055a>)
- Check that the changelog file mentions your login name as release manager for this release (e.g. "Tomcat 9.0.94 (markt)").
- Check whether the KEYS file differs from <https://dist.apache.org/repos/dist/release/tomcat/tomcat-9/KEYS> one. The latter one will be replaced after you do a release. Check that the KEYS file contains your public key.
- Check that the full build still works (ant release, preferably). Check that Buildbot builds (<https://tomcat.apache.org/ci.html#Buildbot>) are green.
- Prepare your build environment:
  - Add a build.properties file (or, better yet, keep this file in your ~/build.properties if you have a dedicated build machine) with the following configuration (adjust paths for your environment)

```
execute.validate=true

execute.test.bio=true
execute.test.nio=true
execute.test.apr=true

test.haltOnFailure=true

gpg.exec=C:/Program Files (x86)/GNU/GnuPG/gpg2.exe
#gpg.exec=/usr/bin/gpg

base.path=C:/temp/libs
#base.path=~/tomcat-libs
# Enable the following if the DigiCert ONE magic is all set up, including ~/.digicertone
/pkcs11properties.cfg
codesigning.storepass=apikey|keystorepath|keystorepassword
do.codesigning=true
```

### Build the release + create git tag

The aim is to create a copy of the current trunk but without the "-dev" appended to the end of the version number. All artifacts required for repeatable builds will be included as well.

- Perform an git clone and switch to the correct branch
  - `git clone https://github.com/apache/tomcat /c/releases/asf-tomcat` or `git clone -b 9.0.x git clone https://github.com/apache/tomcat /c/releases/asf-tomcat`
  - `cd /c/releases/asf-tomcat`
  - `git checkout 9.0.x`
  - `git pull`
- (Hint: using `ant pre-release` may save you the following step, plus steps 2-3 in the Maven release process.)
- Edit "build.properties.default" and change the lines after "# ----- Reproducible builds -----" to a new value. (NOTE: this is done by 'ant pre-release')
  - Note that the value of `ant.timestamp.now` property is in seconds (unlike the value returned by `System.currentTimeMillis()` method which is milliseconds, see [bug 65527](#) for how this happened for Tomcat 8.5.70). The value can be printed in a Bash shell with the following command:  
`date +%s`  
To print seconds since epoch, and date and time in human-readable format in UTC time zone:  
`date -u '+%s %Y-%m-%d %H:%M:%S %Z'`
- Edit "build.properties.default" and change the line `version.dev=-dev` to `version.dev=` (NOTE: this is done by 'ant pre-release')
  - This can be done with `sed -i.bak "s/^version.dev=.*\/version.dev=/" build.properties.default`
- Edit "webapps/docs/changelog.xml" and replace `rtxt="in development"` with `rtxt="Release in progress"` for the version being released (NOTE: this is done by 'ant pre-release')
  - Remember that `notepad.exe` will remove UTF-8 Byte-Order Marks (BOMs). Use `write.exe` or `Notepad++` instead.
- Build the release
  - `ant release`

- Commit these changes
  - `git add --all` (to pick-up the repeatable build artifacts in addition to the content-changes from above)
  - `git commit --all --message "Tag 9.0.94"`
  - `git tag 9.0.94`
  - `git push origin 9.0.94`
- Check the diff mailed to the dev list

Notes:

- GPG should be configured to use your Apache code signing key by default
- I always ensured `c:/temp/libs` was empty so that the build had to download all the dependencies
- This does not include the signing of the Windows installer, which must be done using <https://one.digicert.com/> and <https://infra.apache.org/digicert-use.html> which is automated during the build process once the Tomcat PMC key is accessible by using `jsign`. (Working! Try setting `do.codesigning=true` in `build.properties`).
- The logs for the Windows signing are in `~/signingmanager/logs` on Linux
- If you get an error signing the installer/uninstaller "Cannot load keystore `~/digicertone/pkcs11properties.cfg`" then you probably don't have your environment variables (or `codesigning.storepass` property) set properly.

## Upload the release

Upload the contents of `TOMCAT_9_0_XX/output/release` to <https://dist.apache.org/repos/dist/dev/tomcat/tomcat-9/>

- `svn checkout https://dist.apache.org/repos/dist/dev/tomcat/tomcat-9`
- `cp -r TOMCAT_9_0_XX/output/release/* tomcat-9`
- `cd tomcat-9 && svn add v* && svn commit -m "Upload vX.Y.Z for voting"`

## Generate the Maven artifacts

See <https://github.com/apache/tomcat/blob/9.0.x/res/maven/README.txt> steps 1 to 3 for the release (not step 4 until the vote passes !)

## Call a vote

- E.g. <http://markmail.org/message/gvmbwocspnwb2dfe>
- Increment the version number ready for the next tag (e.g. <https://svn.apache.org/viewvc?view=revision&revision=1500094>)

## Cleaning Up

- Reset to the 9.0.x branch (`git reset HEAD~ && git checkout build.properties.default && git checkout webapps/docs/changelog.xml && rm -f build.properties.release res/maven/mvn.properties.release res/install-win/Uninstall.exe.sig res/install-win/tomcat-installer.exe.sig`)

I found it simplest to keep this clone for tagging to ensure no other edits found their way into the tag.

To get a clean copy of the release (e.g. for testing):

- `mkdir /c/releases/TOMCAT_9_0_94`
- `git archive 9.0.94 | tar -x -C /c/releases/TOMCAT_9_0_94/`

I'm using Git Bash for the above. Adjust as necessary for you choice of tools.

## If the vote passes

- Send a recap "[VOTE][RESULT]" email to the vote thread on the dev list with the vote counts.
- `svn mv https://dist.apache.org/repos/dist/dev/tomcat/tomcat-9/v9.0.XX https://dist.apache.org/repos/dist/release/tomcat/tomcat-9/v9.0.XX`
- `svn checkout --depth immediates https://dist.apache.org/repos/dist/release/tomcat/tomcat-9/`

and update the `KEYS` file there to be the same as the one used for release. (The download page has links pointing to this file).

- Release the Maven artifacts (<https://github.com/apache/tomcat/blob/9.0.x/res/maven/README.txt> step 4)
- Wait for the release artifacts to appear in these locations:
  - [https://downloads.apache.org/tomcat/tomcat-\[v\]/](https://downloads.apache.org/tomcat/tomcat-[v]/)
  - [https://dldn.apache.org/tomcat/tomcat-\[v\]/](https://dldn.apache.org/tomcat/tomcat-[v]/)

- Update the website (e.g. <https://svn.apache.org/viewvc?view=revision&revision=1901819>)  
 Note: *only* modify the `xdocs/` files, then use `ant docs` to generate the HTML.  
 Note: the index and oldnews pages are sorted by date. Thus the new announcement pops up to the top of the page.  
 Note: using `tools/update-version.sh oldver newver releasedate asfuser` from the 'site' project may help you with almost all of the website updates below.
  - Update the `build.properties.default` file to ensure the correct release is listed.
  - Update `download` page (update "[define v]" macro for the new version)
  - Update `whichversion` page (the current version number).
  - Update `doap_Tomcat.rdf` file (version number and release date).
  - Update `docs/tomcat-[v]-doc/changelog.html` to set the release date to the date the files were copied to the release area in yyyy-MM-dd format ("beta, 2007-02-28" for betas).
  - Update `migration` page (versions in "configuration file differences" comparison form).
    - Notable changes that change behaviour and may require action when upgrading from the previous release may be added to the "noteable changes" section on the `migration` page.
  - Move old announcement to the top of the `oldnews` page.
  - Add announcement for the new release to the top of the front page (`index`).
  - Update the javadocs (see <https://svn.apache.org/repos/asf/tomcat/site/trunk/README.txt>)
- Announce the release (e.g. <http://markmail.org/message/xyantb3ozzmucdjt>) to `users@t.a.o`, `cc dev@t.a.o`, `announce@t.a.o`, [announce@a.o](mailto:announce@a.o) (try `. /tools/update-version.sh oldver newver date asfuser email` where "email" is literal, to auto-generate your announcement email message)
- Add the version number in Bugzilla (<https://bz.apache.org/bugzilla/> Administration Products / versions)
- Drop the artifacts for the previous release from [https://dist.apache.org/repos/dist/release/tomcat/tomcat-\[v\]/](https://dist.apache.org/repos/dist/release/tomcat/tomcat-[v]/)
- Add release data (version and date) to `reporter.apache.org` - <https://reporter.apache.org/addrelease.html?tomcat>

## If the vote does not pass

- Update the release date in the changelog to "not released"
- Drop the artifacts from <https://dist.apache.org/repos/dist/dev/tomcat/tomcat-9/>
- Drop the maven artifacts from the staged maven repository