Verifying a Release Build

Recent (c. fall of 2022 and later) Apache Tomcat releases should be fully-reproducible as long as you follow some rules. This page describes how to perform a verification of a release. This process can be performed by *anybody, anywhere, at any time*. It's not just something that Tomcat committers, PMC members, or release-managers can do.

We encourage the community at large to verify these builds themselves to ensure we are doing them properly and that they have not been tampered with in any way. You should be able to verify a proposed release during the *voting* period for that release, or verify a released version once the voting has ended (and the vote has passed).

Verifiable Items

All release artifacts should be verifiable. That includes `apache-tomcat-*.zip`, `apache-tomcat-*.tar.gz`, in both source and binary distributions, as well as the Windows .exe installer and uninstaller binaries.

Prerequisites

There are several things which you will need in order to perform a Tomcat release build.

1. A build environment running on x86 or x86-64 architecture. This is required to build the Windows installer and uninstaller binaries which require NSIS which is an x86-only product. Use of Microsoft Windows is not required; builds can be completed on MacOS or Linux with wine installed. Please see Tomcat's Release Process for how to install and configure wine if you choose this option.
2. A Java Development Kit (JDK) which matches the version used to build the release. This version information can be found in each release's source artifact in the build.properties.release file.
3. Apache Ant which matches the version used to build the release. This version information can be found in each release's source artifact in the build.properties.release file.
4. GnuPG and your own private key. You will be asked to sign the release that you build locally during the build process.

You will need to configure your build environment. The easiest way to do that is to create a `build.properties` file in your user's home directory which contains the following configuration:

```
gpg.exec=(full path to gpg.exe on your system)
base.path=(full path to a temporary directory where Tomcat can download dependencies e.g. /tmp/tomcat-build-libs)
```

How to Verify

You will need to download the source distribution of the release. This is the apache-tomcat-x.y.z-src.zip or apache-tomcat-x.y.z-src.tar.gz file which you can find in the downloads area for Apache Tomcat. For example, the source ZIP file for Apache Tomcat 10.1.5 can be found at https://dist.apache.org/repos/dist/dev/tomcat/tomcat-10/v10.1.5/src/apache-tomcat-10.1.5-src.zip

Expand the ZIP or tar.gz archive and `cd` into that directory. Inspect the `build.properties.release` file to ensure you are using the same versions of both the JDK and Apache Ant.

Perform a release build:

```
ant release
```

This will take some time, but when it has completed you should have a number of release artifacts in the `output/release` directory. These are the artifacts you will be verifying.

To verify release artifact files, you will need to download those files you wish to verify. You have already downloaded the source artifact, so that one is very easy to verify:

```
diff output/release/apache-tomcat-x.y.z-src.zip ../apache-tomcat-x.y.z-src.zip
```

Or

```
diff output/release/apache-tomcat-x.y.z-src.tar.gz ../apache-tomcat-x.y.z-src.tar.gz
```

If you have downloaded all the release artifacts to a directory called `release-verification` then you should be able to run this command on UNIX-like systems (or in Windows using bash):

```
(cd output/bin ; find . \( -name "*.asc" -prune \) -o \( -type f -exec diff -qs "/path/to/release-verification/
{}" "{}" \; \) )
(cd output/src ; find . \( -name "*.asc" -prune \) -o \( -type f -exec diff -qs "/path/to/release-verification/
{}" "{}" \; \) )
```

This will print out the names of all files compared and whether or not they match each other. We skip the `.asc` files because those were produced by the release manager using their private GPG key and will not be reproducible by you.

How does all this work?

When the release-manager performs the release, all parts of the build are tweaked so that reproducibility is possible. There is support from Apache Ant, the javac compiler, and other parts of the toolchain to ensure that every file generated during the build is identical.

This begins by using the same versions of the toolchain that were used by the release-manager. It's possible that two different versions of e.g. javac willl generate different output, so it's necessary to use the same versions for verification.

The second step is found within `build.properties.release` which contains the timestamp that the release was prepared by the release-manager. This ensures that all files generated during the build process have consistent file timestamps which is especially important when they are put into archives which store those file timestamps.

The third step is with the file signature-generation. Apache Tomcat releases have several different types of file signatures:

- Simple hashes, such as files with names ending in `.sha512` . These are directly reproducible by re-hashing the files once built, and verifiable with `diff` .
- GPG signatures, which are not reproducible but still verifiable with `gpg --verify` .
- Windows .exe digital signatures, which are the most complicated. The release-manager generates so-called detached signatures (because the file contains the signature only) and bundles these in the release tag in revision-control, as well as in the source release. You got a copy of these files when you expanded the source archive. When *you* perform a release-build, the release process uses the existing detached signature files to merge with the unsigned .exe files and the result is a byte-for-byte copy of the signed installer and uninstalled binaries. You can verify these with d `iff` or by asking Windows to confirm the digital signature of the files.

Can I do this more easily?

Yes, if you trust that the release itself comes with trustworthy tools.

In Apache Tomcat releases starting (with luck!) with 8.5.98, 9.0.85, 10.1.18, and 11.0.0-M16, you can have an automated release-verification process.

First, download the source distribution of the release, unpack it, ensure you have the required build environment for your release (mostly a proper JDK and Apache ant version), then run:

```
ant release

ant verify-release
```