Log4j Kotlin API Release Process

How to perform a release of the Log4j Kotlin API.

Step-by-step guide

Prerequisite Steps

1. Run `mvn clean verify` on the project to make sure it builds cleanly.
2. Make sure you have an up to date PGP key (at least 4096-bit RSA or stronger; use of ECDSA or EdDSA keys should be in addition to the RSA keys for broader compatibility) in the Logging KEYS file and having it mirrored in a common public PGP key repository such as Ubuntu Keyservers.
3. Configure your `git config user.signingKey`, `user.name`, and `user.email` values to match that key.
4. Ensure you have your credentials set up in `~/.m2/settings.xml` as described in Log4j 2 Release Process.
   a. TODO: insert information about using encrypted passwords.

Release Steps

1. Edit `pom.xml` and change the `Log4jKotlinVersion` property to the new version n.n.n.
2. Update the `Log4jReleaseManager` and `Log4jReleaseKey` properties if necessary.
3. Generate the site locally and verify it looks alright.
   a. `mvn site && mvn site:stage -DstagingDirectory=$HOME/log4j-kotlin`
4. Build, sign, tag, and publish a release candidate n.n.n-rcn. If your GPG key isn't on a hardware key, it should be password protected; you'll be prompted to enter the password to sign artifacts during the build.
5. Login to https://repository.apache.org/ with your ASF credentials. Select "Staging Repositories", then find the newly created orgapachelogging-* staging repository and close it.
6. Checkout the release tag and stage the website into a working copy of the asf-staging branch of https://github.com/apache/logging-log4j-site/tree/asf-staging to log4j-kotlin-n.n.n and update the symlink of kotlin to point to it.
   a. `mvn site && mvn site:stage -DstagingDirectory=$HOME/code/logging/site-java/log4j-kotlin-n.n.n`
   b. Check result of deployment: https://logging.staged.apache.org/log4j/kotlin/
8. Run for f in *.bz2 *.gz *.zip; do sha256 -a 512 $f >$f.sha512; done to generate hashsums.
9. Send a release vote email to dev@logging.apache.org with links to everything.
10. If the release fails, drop the staging repository on repository.a.o and start over with a fresh release candidate number.
11. Otherwise, if the release passes, complete the release on the staging repository.
12. Create an immutable signed release tag.
   a. `git tag -s rel/n.n.n 'log4j-api-kotlin-n.n.n-rcn'` -m 'Release n.n.n of Log4j Kotlin API'
   b. `git push --tags`
   a. `svn mv $(svn ls -r dev/logging/log4j/kotlin/*) ^/release/logging/log4j/kotlin/`
   b. `svn mv $(svn ls -r dev/logging/log4j/kotlin/ | xargs printf '%s') ^/release/logging/log4j/kotlin/`
14. Login to https://reporter.apache.org/ to record the release.
15. Wait 12-24 hours for mirrors to propagate.
   a. `svn rm ^/release/logging/log4j/kotlin/n.n.n-1`
17. Merge the asf-staging site branch into asf-site
   a. `git checkout asf-site && git rebase asf-staging && git push origin asf-site`
18. Update any redirect to point to new version for latest (if applicable).
19. Send announcement email (from your apache.org email) to dev@logging.apache.org, log4j-user@logging.apache.org, and announcements@apache.org.

Related articles

- Log4j 2 Release Process
- Log4j Kotlin API Release Process
- Logging Parent POM Release Guide
- Log4j Scala API Release Process