Log4j 2 Release Process

Before beginning your first release, take the following preparatory steps:

1. Run `mvn clean install` on the project to make sure it will build cleanly.
2. Run `mvn -P rat -DskipTests clean verify` to verify all the licenses are valid.
3. Login to `http://repository.apache.org` using your ASF credentials. Click on your username in the upper-right-hand corner and click Profile. Make sure your email address is correct. If not, contact Infra to get it corrected. You will be unable to get the necessary notifications until it's corrected.
4. Make sure you have a 4096-bit RSA PGP key pair for code signing. The public key should be published in a public repository, preferably `http://pgp.mit.edu/`. Also make sure the public key is published to `http://www.apache.org/dist/logging/KEYS`.
5. Edit `~/.m2/settings.xml` to add the corresponding private key and its passphrase to your settings. Also add the Apache Subversion server credentials and Sonatype repository credentials to your settings (this is your ASF credentials). You'll want to blank out the passphrase after each release and fill it back in prior to each release so that it isn't compromised.

```xml
<profiles>
  <profile>
    <id>apache-release</id>
    <properties>
      <gpg.keyname>ABC123FC</gpg.keyname>
      <gpg.passphrase>thisIsMyPassphrase</gpg.passphrase>
      <gpg.executable>gpg2</gpg.executable><!-- Use this if executable differs from "gpg" -->
    </properties>
  </profile>
</profiles>

<activeProfiles>
  <activeProfile>apache-release</activeProfile>
</activeProfiles>
...
<servers>
  <server>
    <id>svn.apache.org</id>
    <username>myUsername</username>
    <password>myPassword</password>
  </server>
  <server>
    <id>apache.releases.https</id>
    <username>myUsername</username>
    <password>myPassword</password>
  </server>
  <server>
    <id>apache.snapshots.https</id>
    <username>myUsername</username>
    <password>myPassword</password>
  </server>
</servers>
```

Once you've done this, take the following steps to perform a release:

1. Edit pom.xml and change the Log4jReleaseVersion property to the version you are releasing.
3. Update src/changes/announcement.vm
4. Run `mvn -P release-notes generate-resources` to create the release notes.
5. Update src/changes/changes.xml with the release date.
6. Pick a logo from src/site/resources/logos. Edit it with Gimp to add the release version number and save it as a png. Copy the saved logo to src/site/resources/images/logo.png.
7. Run git add, git commit, and git push to commit the RELEASE-NOTES.md file that was just generated along with the other files that were modified.
8. Run `mvn site -Dhttps.protocols="TLSv1.2"` followed by `mvn site:stage -DstagingDirectory=$HOME/log4j` where $HOME is your home directory and verify that the site looks good.
10. Login to `http://repository.apache.org` using your ASF credentials. Select “Staging Repositories” then check the org.apache.logo repository and close it.
11. Check out the release tag that was created via the Maven release plugin using git checkout tags/tagname.
12. Run `mvn site` in the tag directory. When that completes run `mvn site:stage -DstagingDirectory=$HOME/log4j` where $HOME is your home directory.

13. Zip the contents of $HOME/log4j with `zip -r log4j.zip log4j`. Commit the site to your github.io account.


15. Add the site zip to where https://dist.apache.org/repos/dist/dev/logging/log4j/ is checked out on your machine.
   a. Sign the site ZIP file using `gpg --armor --output apache-log4j-2.x-site.zip.asc --detach-sig apache-log4j-2.x-site.zip`. You can use the `--local-user [key ID]` option (it must come before `--detach-sig`) to specify the exact key to use when signing the ZIP file.

16. Generate the release vote email:
   a. Create the email addressed to dev@logging.apache.org. Sending the email to the PMC is not necessary.
   b. Copy the changes in the release from RELEASE-NOTES.txt into the email.
   c. Provide a link to the tag, web site on http://people.apache.org, and the artifacts in the Nexus repository.
   d. Provide the command to download all the artifacts: `wget -e robots=off --cut-dirs=3 -r -p -np --no-check-certificate $LINK` where $LINK is the URL to the repository you just closed (plus the org/apache/logging/log4j/ path appended).

17. If the release vote fails proceed as described in the section below, otherwise if it passes:
   a. Create a new tag named log4j-n.n from the log4j-n.n-rcn tag by changing to the directory containing the source for the release and performing `git tag -a log4j-n.n -m "Release n.n of Log4j"` followed by `git push --tags`.

18. The following steps must be performed by a PMC member:
   b. Create the release directory under the log4j directory.
   c. Move all the distribution artifacts from the distribution dev location to that directory.
   d. Perform an `svn add` of the release directory.
   e. Commit the release to Subversion.
   f. Login to reporter.apache.org and add the release version and date.

19. The following steps can then be taken by whomever started the release:
   a. Delete the core-its project from the Nexus repository.
   b. Release the remaining artifacts in the Nexus Repository.

20. Wait 12-24 hours (or after the distribution artifacts have been propagated to the mirrors and the Maven artifacts have been propagated to the Central Repository), and then the following steps must be performed by a PMC member:
   a. Perform `svn delete` on the previous release directory for the same version under the log4j directory (https://dist.apache.org/repos/dist/releases/logging/log4j).
   b. Commit the delete to Subversion.
   c. See ManagingTheWebSite regarding sub-projects.
   d. Copy the release web site from the local directory where it was deployed into the sub-project directory and make it the current link.

21. After the website is updated, send the release announcement email. This should be sent out to dev@logging.apache.org, log4j-user@logging.apache.org, private@logging.apache.org, and general@logging.apache.org.

22. Send the release announcement to announcements@apache.org using your apache.org email address.

If the release fails before sending the vote email:

1. Login to http://repository.apache.org using your ASF credentials. Select "Staging Repositories" then check the org.apache.logging repository and drop it.
2. Revert any changes that have not been committed.
3. Restart the release process as the same release candidate.

If the release fails after sending the vote email:

1. Login to http://repository.apache.org using your ASF credentials. Select "Staging Repositories" then check the org.apache.logging repository and drop it.
2. Rename the release tag in Git to add rcn to the end of the tag. Note that this may not be possible in Git depending on configuration.
3. Restart the release process as a new release candidate.