Log4j Scala API Release Process

Release Parameters

When preparing a release candidate, there are a few parameters either prompted for or needed to be updated manually. These include:

- Tag version: e.g., `12.3-rc4`
- Release version: e.g., `12.3`
- GPG key id: e.g., `1234567887654321`
- Log4j Core dependency version (both the published dependency and documentation example)

Preparation

1. Verify the project builds cleanly via `./sbt "+test"`
2. Verify the site builds cleanly via `./sbt makeSite`
3. Verify the files pass the audit check via `./sbt auditCheck`
4. Make sure you have a 4096-bit RSA PGP key for signing releases uploaded to a public PGP key repository (preferably http://keyserver.ubuntu.com) as well as located in the KEYS file.
5. Configure sbt with your pgp key by first exporting into the old format for use by sbt-pgp by running `gpg --export-secret-keys --armor --output ~/.sbt/gpg/secring.asc` and then edit the file `~/.sbt/gpg.sbt` with the following:

   gpg.sbt

   ```
   useGpg := true
   pgpSecretRing := Path.userHome / ".sbt" / "gpg" / "secring.asc"
   usePgpKeyHex("1234567887654321")
   ```

6. Configure your Apache credentials by writing to `~/.sbt/1.0/global.sbt` with something like:

   global.sbt

   ```
   credentials += Credentials("Sonatype Nexus Repository Manager", "repository.apache.org", "username",
   "password")
   ```

Releasing

1. Run `./sbt release` and answer the version number prompts.
   a. **Important:** the version number used here should be a release version like `12.3` due to script limitations. When prompted to push your local changes, say "No", and move the tag: `git tag v12.3-rc3 'v12.3^{}' && git tag -d v12.3 && git push --tags`
   b. Close the staging release repository afterwards by logging in to https://repository.apache.org/#stagingRepositories and finding the appropriate `orgapachelogging-NNNN` repository.
2. Check out the created tag and run `GPG_KEYID=mykeyid ./create-distributions.sh <release-version-number>`
   a. *FIXME:* binary distributions may not be signed by default; make sure to sign them! As a workaround, use the following bash script: `cd target; for f in *.zip *.gz; do shasum -a 256 $f >$f.sha256; shasum -a 512 $f >$f.sha512; done`
3. Commit those to https://dist.apache.org/repos/dist/dev/logging/log4j/scala for staging (removing previous release candidates as necessary).
4. Commit the site to github pages for previewing.
   a. *FIXME:* this currently requires manually aggregating the API docs from each `target/scala-2.xx/api/` directory into `target/site/api/2.xx/`. Can be done via: `cd target/site/api; mv ../../scala-2.10/api 2.10` ...
5. Perform a release email and follow the usual release process as Log4j Core does.