

Validate a Release

Validating an Apache jclouds release involves verifying the following:

- Verify that the checksums are valid.
- Verify that the PGP signatures are valid.
- Check that the expanded source archive matches contents of RC tag.
- Verify that the expanded source archive builds and passes tests.
- Check that LICENSE and NOTICE files are present and correct.
- Make sure all files have license headers where appropriate.
- Check that all dependencies have compatible licenses.
- Verify that no compiled archives bundled in source archive.

Some steps require a manual verification, and others are fully automated. The following scripts can be used:

Verify RAT, build, tests, checksums and signatures in one script

1. Download the verification script:
 - a. Unix: https://dist.apache.org/repos/dist/dev/jclouds/verify_jclouds_rc.sh (if you're running this on a Mac, you'll need [brew](#) and to do a brew install gpg first).
 - b. Windows: https://dist.apache.org/repos/dist/dev/jclouds/verify_jclouds_rc.cmd (Set the values of the variables TAR_EXEC ([TarTool](#)), FCI_V_EXEC ([FCIV](#)), GPG_EXEC and SED_EXEC ([GNU sed for Windows](#)) to match your local environment).
2. Run it and watch for failures:
 - a. Unix: `./verify_jclouds_rc.sh 1.9.2-rc1`
 - b. Windows: `.\verify_jclouds_rc.cmd <dist-folder> 1.9.2-rc1`

Check that the expanded source archive matches contents of RC tag.

1. Download the tag comparison script: https://dist.apache.org/repos/dist/dev/jclouds/compare_tarball_tag.sh
2. Run it as: `./compare_tarball_tag.sh <path to the folder where all the tar files are>.`
3. Watch for differences.

Validate the Maven staging repos

On your local machine create a directory and a pom.xml file to download the release candidate JARs from the staging repo. In the [VOTE] thread look for the "Maven staging repos" section and replace the xxxx below with the value from the "Maven staging repos" section.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <properties>
    <jclouds.version>1.6.3</jclouds.version>
  </properties>
  <groupId>org.apache.jclouds</groupId>
  <artifactId>jclouds-staging</artifactId>
  <version>1.0</version>
  <dependencies>
    <dependency>
      <groupId>org.apache.jclouds</groupId>
      <artifactId>jclouds-all</artifactId>
      <version>${jclouds.version}</version>
    </dependency>
    <!-- Add any jclouds additional dependency here -->
  </dependencies>
  <profiles>
    <profile>
      <id>jclouds-staging</id>
      <repositories>
        <repository>
          <id>jclouds-staging</id>
          <url>https://repository.apache.org/content/repositories/orgapachejclouds-xxxx/</url>
        </repository>
      </repositories>
    </profile>
  </profiles>
</project>
```

Run the command `mvn clean -Pjclouds-staging dependency:copy-dependencies "-DoutputDirectory=./lib" -U` to download the JARs to a lib directory. At the end of it you should have a simple directory structure like so.

```
jclouds-1.9.2$ ls  
lib/      pom.xml
```

With the lib directory full of jclouds JARs. You can now use these JARs to do things like run the [jclouds-examples](#). If something goes wrong, notify the dev list.