2016-03 How to Review a Release and Vote

Introduction

This document is for software release *reviewers* and describes **how to review** an Apache Taverna incubator release, not **how to prepare** the release candidate.

Permalink: https://s.apache.org/review-release

Contents

NOTE: In this document, items from the ASF release checklist are identified by (RC) and items from the ReleaseChecklist wiki are identified by (RCw).

I received a VOTE email, what do I do now?

Congratulations! You are about to participate in a important Apache Software Foundation activity. This document focuses on what is required to review a Taverna release. Details: How to Review A Release will walk you through the steps in more detail, if necessary.

Reviewer's tasks

Reviewers must:

- **Download the source artifact and check it.**
  - Download the release candidate source artifacts from the VOTE email.
  - See download instructions in the README file (example).
  - See the following sections for information about **what** to check and **how** to check it.
  - For more information see Details: How to Review A Release
- **Support the release manager**
  - Report on your progress / success
  - Raise any issues on the DISCUSS thread

Tools

Some necessary and optional tools:

- GPG (required)
- MD5 and SHA Checksum utility (optional) [download]
- Text file difference checker (optional) [online]

Before you start

Is there a reason *not* to do the release?

Speak up immediately if you know of a reason not to do the release. For example, if there is a critical security bugfix that has not been included, alert the community so it can be fixed before the release.

Each project community determines its release philosophy. Projects that "release early, release often." like Taverna, will likely allow certain bugfixes to wait until the next release. See the guidelines for Taverna's release philosophy.

Are the prerequisites installed?

The README.md file for each distribution lists the prerequisites, such as which Java or Maven versions are required. [example]

Download the release candidate(s)

Download the release candidate(s), including all hash and signature files, using the links in the VOTE email. (The release candidates are Zip files, the hash files are .md5 and .sha1 files, and the signature files are .asc files.)

What to check

Check the following items (in order).

1. Checksums and PGP signatures are valid (RC)

Check that the MD5 and SHA checksums of the downloaded release candidate match the values in the VOTE email.

Use the Apache Taverna KEYS files to check that the signature is valid.

(See the Release Signing dev documentation. (RC))

2. Commit ID matches value in VOTE email.

Check that the git commit ID of each distribution matches the value in the VOTE email.

3. Disclaimer is correct and file names include "incubating" (RC)

Disclaimer

- Verify the incubator disclaimer is in a separate file called DISCLAIMER, residing inside the top-level distribution folder, along with the LICENSE and NOTICE files.
- Verify the DISCLAIMER file text matches that in the Podling Branding Guide.

File names: Verify that all file names include "incubating."

4. Top-level LICENSE and NOTICE are correct for each distribution (RC)

Verify the top-level LICENSE and NOTICE files in the distribution match ASF guidelines, plus any Taverna-specific requirements.

See Details: How to Review A Release for more information.

(Also, see the Licensing How-To, plus various pages under Legal Affairs. (RC))

5. All source files have license headers, where appropriate (RC)

- Make sure code that is (1) developed at the ASF or (2) developed elsewhere **and submitted by the copyright owner or owner's agent** have the appropriate source file headers.
- Check that all other source files (AKA third party files) have been handled properly.

(See the ASF Source Header and Copyright Notice Policy. (RC))

6. The provenance of all source files is clear (ASF or software grants). (RC)

(See the IP clearance section of the Mentor's guide, as well as the Releases section of the Incubator's policy page.} (RC)

This is generally relevant where we have accepted a new larger contribution, e.g. a GSOC student has contributed a new module.

7. Dependencies licenses are ok as per http://apache.org/legal/ (RC)

Check that all dependency licenses have been handled correctly and that no Category X licenses have inadvertently been included.

8. Release consists of source code only, no binaries. (RC)

NOTE: CHECK THIS BEFORE YOU BUILD.

Each Apache release must contain a source package. This package may not contain compiled components (such as "jar" files) because compiled components are not open source, even if they were built from open source.

"The source artifact is the thing being released. Binaries and git are secondary."

9. Build is successful, including automated tests (RC)

The expanded source archive is expected to build and pass tests. The goal is to receive a BUILD SUCCESS message at the end of the building and testing process.

Do not skip any automated unit tests (E.g., do not use -DskipTests=true)

10. Verify the source produces the correct binaries

In addition to cursory checks of the jars, **at least one person should check** that all staged JARs are the same as those built from the downloaded release candidate. One approach is to do a recursive **wget** of the repository , and then compare the result of "**find . -name '*jar'**" in the wget-tree with ***/*/target/*.jar**.

NOTE: Binary releases are considered "convenience only" and are not crucial for the vote: The source release is what everything else should be made from. However, in practical terms, most people download the binaries from the Maven repository. Therefore, it is important this is checked at least once.

How to check

Detailed Instructions for Reviewing a Release contains detailed instructions for how to check (verify) the release.

Voting and community guidelines for acceptance

General review and voting requirements

- Minimum vote: The minimum requirement is **three +1 votes** with a majority in favour.
- Comments: The release manager decides how to handle comments.
- Quality: Does the release quality level meet the group norms? (For example, "can we live with it?" vs "is it perfect?")

Minimum Taverna release review guidelines (TBR)

This is a draft list of candidate community guidelines. It has not yet been validated by the community.

ALL

- Download *at least one* distribution (source-release-zip) and ensure it builds successfully
- Verify checksums and signatures

PPMC Members (and others, if they want)

- Ensure accuracy of the following
  - Top-level LICENSE and NOTICE files
  - Source file headers ("Apache" and 3rd party headers)
  - Dependency licenses
  - Source archive (does not include any binary files)
- Verify commit ID **(At least one PPMC member)**

NOTE: Functional testing will be limited until the full Taverna suite has been released.

(See also Podling Releases and voting process.)

Possible additional items to check

A list of possible additional items is maintained on the ReleaseChecklist wiki page. This is the list (as of 3/2016):

- Provide build instructions, unless obvious. (RCw) *(obvious to whom?)*
- Match each source archive *(define)* with the corresponding SCM *(define)* tag. (RCw)
- Ensure RAT *(link? define?)* report is clean. (RCw)
- Ensure change log is clean. (RCw) *("clean" is determined by each project? replace with "meets the community guidelines"?)* - this appears to be referring to release notes.
- Ensure all copyright dates are current. (RCw)
- Ensure issue tracker (e.g., JIRA) is clean. (RCw)
- Run extended tests (if any) and ensure they pass.(RCw)
- Test that build succeeds on all target platforms.(RCw)
- Ensure documentation builds correctly.(RCw)
- Ensure binary release does not contain redundant dependency archives.(RCw)

Definitions

- **binary files**: Created during *mvn clean install,* located in target folders. Includes pictures, ZIP files, and JAR files.
- **dependency**: "A dependency is a file that something you are trying to install requires." [AskUbuntu]
- **distribution:** A distribution is all of the files (not including dependencies) that are needed to run an application.
- **license**: Terms and conditions for use of source code. For example, *Licensed under a Creative Commons Attribution 3.0 license.*
- **notice**: Copyright notice. For example, *Copyright (c) 2012-2015 University of Manchester.*
- **provenance**: "[A] record that describes the people, institutions, entities, and activities involved in producing, influencing, or delivering a piece of data or a thing." (Schreiber, 2013)

- **source artifact**: ".. 'things' ... produced by people involved in the process. Examples [are] design documents, data models, workflow diagrams, test matrices and plans, setup scripts. ... *[A]ny* thing that is created could be an artifact." [Programmers StackExchange]
- **source files**: Files downloaded from the VOTE email. Includes *.java and *xsd.

References:

Schreiber, Andreas. (2013) "**Increasing software quality using the provenance of software development processes**," *in ESA Software Product Assurance Workshop 2013, 12-13 June 2013, Noordwijk, Niederlande.* [link]